



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um Estudo sobre o Valor do Trabalho no Desenvolvimento de Software

Ícaro Eustáquio Nunes de Souza

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora

Prof.^a Dr.^a Maria de Fátima Ramos Brandão

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Henrique Veneziano

Banca examinadora composta por:

Prof.^a Dr.^a Maria de Fátima Ramos Brandão (Orientadora) — CIC/UnB
Prof. Dr. Perci Coelho de Souza — IH/UnB
Prof. Dr. Francisco Villa Ulhôa Botelho — CIC/UnB

CIP — Catalogação Internacional na Publicação

Souza, Ícaro Eustáquio Nunes de.

Um Estudo sobre o Valor do Trabalho no Desenvolvimento de Software
/ Ícaro Eustáquio Nunes de Souza. Brasília : UnB, 2015.

99 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. Métrica de Software, 2. Valor do Trabalho em Software, 3. Trabalho Imaterial, 4. Sociedade Informacional, 5. Teoria Crítica do Valor do Trabalho, 6. Sociologia do Trabalho

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Um Estudo sobre o Valor do Trabalho no Desenvolvimento de Software

Ícaro Eustáquio Nunes de Souza

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof.^a Dr.^a Maria de Fátima Ramos Brandão (Orientadora)
CIC/UnB

Prof. Dr. Perci Coelho de Souza Prof. Dr. Francisco Villa Ulhôa Botelho
IH/UnB CIC/UnB

Prof. Dr. Wilson Henrique Veneziano
Coordenador do Curso de Computação — Licenciatura

Brasília, 04 de Setembro de 2015

Dedicatória

Dedico este trabalho a Deus, minha família, amigos que nunca deixaram de acreditar em mim, e a todos os alunos que em sua vida acadêmica já passaram por dificuldades, mas não desistiram de seus sonhos.

Agradecimentos

Agradeço à Deus que em sua infinita misericórdia me cumulou de graças e bênçãos, e que esteve comigo em todos os momentos tristes e alegres ao longo dessa vida acadêmica.

À Universidade de Brasília, que além da formação acadêmica, foi fundamental na minha formação como pessoa. Onde aprendi que a vida tem altos e baixos e não se deve desistir nunca. A todos os professores que tive contato e participaram diretamente neste processo de formação acadêmica e pessoal, meu muito obrigado.

À minha família que viveu ao meu lado dando forças, tendo paciência e amor comigo. A meus pais pelo dom da vida e doação incondicional desde os primeiros passos até a conclusão dessa etapa. Às minhas irmãs, que em muitos momentos viveram ao meu lado todos as nuances de sentimentos e emoções que a Universidade nos proporciona, e que sempre acreditavam e me ajudaram a superar as dificuldades e lentidão, mas que hoje comemoram comigo este momento de graça e superação.

À minha professora e orientadora Dr.^a Maria de Fátima Ramos Brandão que enxergou um Ícaro que nem eu mesmo acreditava que existia. Você professora Fátima que teve toda paciência me ajudando a vencer as dificuldades, e apresentar este trabalho de conclusão de curso, que foi além da relação professor aluno para recuperar a autoestima de uma pessoa e ajudar a vencer suas dificuldades. Aos professores Percy e Francisco que aceitaram compor a banca desta monografia e às indicações da literatura para realização da mesma.

A todos meus colegas de faculdade, amigos, em especial, Carlos Rafael e Aloir Martins; e minha namorada, que estiveram ao meu lado me incentivando à todo momento para finalização deste trabalho.

Resumo

Nas últimas décadas, o crescimento da demanda por bens, produtos e serviços de informática tem gerado desafios complexos para as empresas desse setor, bem como, para os trabalhadores e consumidores desse segmento de mercado. Em especial, no setor de software, o valor real do produto ou serviço de software passa por questões de medida que não estão bem resolvidas, seja do ponto de vista conceitual e político, como também, do ponto de vista prático. As discussões variam desde a ideia de que a produção e o mercado de software deve ser livre, aberto e gratuito, até práticas de mercado em que o valor do software é ditado pela lógica da oferta e da procura, como resultado de trabalho imaterial ou intangível e que, portanto, não pode ser mensurado como algo físico ou de esforço que pode ser quantificado e valorado objetivamente. A pesquisa propõe um estudo interdisciplinar nas áreas da Sociologia do Trabalho e da Engenharia de Software para aprofundar o entendimento sobre o valor do trabalho imaterial no processo de desenvolvimento de software, segundo pressupostos da teoria crítica do valor do trabalho. Um Modelo de Valor de Trabalho em Software é proposto e descrito para subsidiar critérios e medidas de valor de software segundo novos paradigmas de sociedade da informação e conhecimento. O método exploratório e de pesquisa documental subsidiou o referido modelo a partir dos fundamentos da Teoria do Valor do Trabalho e da Sociedade Informacional onde foram incorporados estudos dos modelos de desenvolvimento de software em cascata, incremental, de espiral e de desenvolvimento ágil. O modelo propõe cinco dimensões de valor e medida que incluem: Tempo, Esforço, Interações, Conhecimento e Valor social. Como contribuições, o trabalho poderá subsidiar estudos posteriores de aplicação dessas métricas no setor de produção de software nas organizações e os impactos de sua efetividade nas relações de trabalho nesse setor.

Palavras-chave: Métrica de Software, Valor do Trabalho em Software, Trabalho Imaterial, Sociedade Informacional, Teoria Crítica do Valor do Trabalho, Sociologia do Trabalho

Abstract

In recent decades, the growth of demand for goods, computer products and services has created complex challenges for companies in this sector, as well as for workers and consumers in this market segment. In particular, in the software industry, the real value of the product or software service goes through issues as they are not well resolved, is conceptually and politically, but also from a practical point of view. Discussions range from the idea that production and software market should be free, open and free until market practices in the software value is dictated by the logic of supply and demand as a result of immaterial labor or intangible and therefore it can not be measured as something physical or effort that can be quantified and valued objectively. The research proposes an interdisciplinary study in the fields of Labor Sociology and Software Engineering to deepen the understanding of the value of immaterial labor in the software development process, according assumptions of critical theory of labor value. A Model Work Value Software is proposed and described to subsidize criteria and software value measures according to new paradigms of society of information and knowledge. The exploratory method and documentary research subsidized said model from the fundamentals of the Theory of Labor Value and Informational Society which were incorporated into studies of software development models cascading, incremental, spiral and agile development. The model proposes five dimensions of value and as include: Time, Effort, Interactions, Knowledge and social value. As contributions, work can support further studies of application of these metrics in the software production sector in organizations and the impacts of their effectiveness in labor relations in this sector.

Keywords: Software Metrics, Software Work Value, Labor Intangible, Informational Society, Critical Theory of Labor Value, Sociology of Work

Sumário

1	Introdução	1
2	O Trabalho na Sociedade da Informação	3
2.1	Teoria do Valor do Trabalho	3
2.1.1	O Trabalho na Concepção Clássica	3
2.1.2	Na Concepção Marxista	5
2.2	Sociedade da Informação	7
2.3	Trabalho Imaterial	11
2.4	Setor de Serviços	13
3	O Desenvolvimento de Software	17
3.1	Processo de Produção de Software	17
3.2	Modelo de Processo de Software	19
3.2.1	Modelo Cascata	20
3.2.2	Modelo Incremental	21
3.2.3	Desenvolvimento em Espiral	24
3.2.4	Metodologia Ágil e o Modelo RUP	25
4	Um modelo de valor do trabalho em software segundo a teoria do valor do trabalho	32
4.1	Princípios da teoria do valor do trabalho no contexto do desenvolvimento de software	32
4.2	Apresentação do modelo de valor do trabalho em software	33
4.3	Métricas de Valor do Trabalho em Software	34
5	Considerações Finais	39
	Referências	41

Lista de Figuras

2.1	Pirâmide de Poderes da Sociedade Informacional - Lojkin (Lojkin, 1995).	9
2.2	Distribuição do Mercado de TI à nível Mundial - (ABES, 2014).	13
2.3	Mercado de TI no Brasil - (ABES, 2014).	14
2.4	Panorama de Serviços no Brasil - (ABES, 2014).	15
2.5	Indicadores de Mercado e Evolução - (ABES, 2014).	15
3.1	Modelo Cascata - Somerville (Somerville, 2007).	22
3.2	Modelo Cascata - Pressman (Pressman, 2010).	22
3.3	Modelo Incremental - Somerville (Somerville, 2007).	23
3.4	Modelo Incremental - Pressman (Pressman, 2010).	23
3.5	Desenvolvimento em Espiral - Somerville (Somerville, 2007).	25
3.6	Visão Geral do RUP	30
4.1	Modelo de Trabalho em Software.	34

Lista de Tabelas

2.1	Confronto entre Revolução Industrial e Revolução Informacional - Lojkine (Lojkine, 1995). Com adaptações	11
3.1	Fatores humanos preponderantes numa equipe de trabalho usando metodologia ágil. Com modificações, Pressman (Pressman, 2010)	28
4.1	Métrica de Trabalho em Software	35

Capítulo 1

Introdução

As atividades de produção e desenvolvimento de software estão relacionadas ao terceiro setor produtivo da sociedade que, ao longo dos últimos anos, tem expandido seu mercado com taxas expressivas de crescimento. Os serviços de informática englobam atividades de comércio de produtos e serviços de bens intangíveis para outros negócios e para consumidores finais. No Brasil, o setor de informática foi responsável por 2,6% do PIB brasileiro em 2014, refletindo uma movimentação financeira de aproximadamente 20 milhões de dólares, em serviços de hardware e software, segundo a Associação Brasileira das Empresas de Software (ABES, 2014).

A demanda crescente de produtos e serviços de informática torna esse setor competitivo e com novos desafios para as empresas. Os requisitos de qualidade e de diferenciação dos produtos e serviços são exigidos nos ambientes das empresas que, por sua vez, buscam funcionários capacitados, criativos e comprometidos na tentativa de criar soluções de sucesso para atender as expectativas dos clientes. Nessa perspectiva, o setor de serviços aposta na interação entre as pessoas e na relação com o cliente para buscar a inovação e o sucesso. Por outro lado, o viés subjetivo da relação de satisfação do cliente, aliado ao consumo de bens e serviços intangíveis em software, torna a aferição de valor uma tarefa complexa que necessita ser aprofundada, segundo modelos de medida mais coerentes para os clientes como também para os trabalhadores desse setor.

Em especial, o valor do produto ou serviço de software passa por questões de medida que não estão bem resolvidas, seja do ponto de vista conceitual e político, como também, do ponto de vista prático. As discussões variam desde a ideia de que a produção e o mercado de software deve ser livre, aberto e gratuito, até práticas de mercado em que o valor do software é ditado pela lógica da oferta e da procura, como resultado de trabalho imaterial ou intangível, que não pode ser mensurado como algo físico ou de esforço que pode ser quantificado e valorado objetivamente. Essa questão é ainda mais complexa quando vários trabalhadores estão envolvidos no processo de desenvolvimento de um produto de software sendo recorrente na literatura a ideia de que quanto mais avançado é o sistema tecnológico e as novas formas de organização do trabalho mais alienados são seus trabalhadores (Antunes, 2008). Como questão norteadora da pesquisa o trabalho propõe contribuir com o processo de medida do trabalho no desenvolvimento de software.

Como objetivo geral o trabalho propõe um estudo interdisciplinar nas áreas da Sociologia do Trabalho e da Engenharia de Software para aprofundar o entendimento sobre o

valor do trabalho imaterial no processo de desenvolvimento de software segundo os pressupostos da teoria crítica do valor do trabalho no contexto da sociedade da informação e do conhecimento.

Como objetivo específico, o trabalho propõe um modelo de valor de trabalho para subsidiar critérios e medidas no processo de desenvolvimento de software.

O método exploratório e de pesquisa documental subsidiou o estudo a partir dos fundamentos da Sociologia do Trabalho e da Sociedade Informacional; dos conceitos da Teoria do Valor do Trabalho e do trabalho imaterial; e dos modelos de engenharia de software em cascata, incremental, de espiral e de desenvolvimento ágil.

O Capítulo 2 apresenta os fundamentos da teoria do valor do trabalho com as discussões atuais sobre o trabalho na sociedade informacional e sobre o setor de serviços de informática. O Capítulo 3 aborda e descreve os principais modelos de processo de desenvolvimento de software: cascata, incremental, espiral e metodologia ágil – o *Rational Unified Process (RUP)*. O Capítulo 4 estabelece as relações da sociologia do trabalho com a engenharia de software para embasamento e proposição de um modelo de valor do trabalho em software para as organizações. O Capítulo 5 tece as considerações finais sobre o estudo e apresenta sugestões de trabalhos futuros.

Capítulo 2

O Trabalho na Sociedade da Informação

Para compreender o atual cenário da computação no ramo de serviços é necessário abordar o tema da Teoria do Valor do Trabalho e outras teorias que foram elaboradas a partir desta ao longo do tempo. Diante disso este capítulo compreende a revisão deste campo da sociologia para buscar um alinhamento com a área da engenharia de software.

2.1 Teoria do Valor do Trabalho

A Teoria do Valor do Trabalho foi desenvolvida no período da Revolução Industrial a partir do século XVIII. É uma teoria social econômica que afirma, em linhas gerais, que o processo de produção tem seu fim num esforço humano. Esta foi descrita por autores como Adam Smith e David Ricardo, e posteriormente trabalhada por Karl Marx, como descrito pelo mesmo em sua obra.

Esses autores, anteriores a Marx, fazem parte dos primórdios do pensamento econômico, escola denominada como - Clássica. Seu pensamento básico é que os mercados se equilibram economicamente num cenário de mudanças econômicas, mas que se finda num longo prazo. Tendo a formulação de sua base teórica através de aproximadamente um século, tendo como expoente os autores já citados acima. Segundo Mattei (Mattei, 2003), os estudiosos da escola clássica acreditavam que a riqueza de uma nação se encontra no trabalho dos cidadãos, e que o valor de troca das mercadorias produzidas são reflexo do esforço do trabalho realizado na sua elaboração.

Doravante para uma melhor compreensão da Teoria do Valor do Trabalho, buscar se a perceber a diferença de pensamento e a mudança do conceito desta teoria ao longo do tempo. Para isso perpassará os três teóricos supracitados, pontuando suas colaborações e formulações, mas também para contrapor as ideias com as de Karl Marx. Tendo em vista que este trabalho buscará desenvolver a teoria sob o viés marxista, e toda sua evolução e atualização ao longo do tempo.

2.1.1 O Trabalho na Concepção Clássica

Considerado por muitos como o “pai da economia política”, Adam Smith através de sua obra *A Riqueza das Nações*, traz alguns conceitos básicos sobre economia política, daí seu apelido, e discorre ainda sobre desenvolvimento econômico e defesa de uma noção de riqueza. A obra deste escritor escocês contém viés Liberal Econômico que tem seu fim

na promoção do bem-estar da sociedade. Mattei (Mattei, 2003) ressalta três aspectos na obra de Smith: a “*riqueza de uma nação*” está no que venha a ser útil e necessário para mesma, mediante o que o trabalho venha proporcionar; a troca é o meio que obtém as mercadorias necessárias; quanto maior e mais eficiente for o trabalho produtivo de uma nação, maior será a acumulação de capital da mesma.

Tendo em vista a importância da figura do trabalho na obra de Smith, fato comprovado quando o mesmo destina falar do tema no início de sua obra. O escritor escocês considera o trabalho fato primordial para a relação de troca entre os homens buscando benefícios para si:

“(...)como é por negociação, por escambo ou por compra que conseguimos uns dos outros a maior parte dos serviços recíprocos de que necessitamos, da mesma forma é essa mesma propensão ou tendência a permutar que originalmente gera a divisão do trabalho.” Smith (Smith, 1996) - *A Riqueza das Nações* - Pg: 74

Logo percebe se a centralidade neste ponto dos temas: trabalho e a troca de mercadorias. A partir destes pilares pode se entender a elaboração da Teoria do Valor do Trabalho pelo autor:

“Portanto, o valor de qualquer mercadoria, para a pessoa que a possui, mas não tenciona usá-la ou consumi-la ela própria, senão trocá-la por outros bens, é igual à quantidade de trabalho que essa mercadoria lhe dá condições de comprar ou comandar. Consequentemente, o trabalho é a medida real do valor de troca de todas as mercadorias.” Smith (Smith, 1996) - *A Riqueza das Nações* - Pg: 87

A partir daí, Mattei (Mattei, 2003) conclui que a teoria do valor, de acordo com Smith, teria uma dupla dimensão: ressaltar o trabalho como medida de riqueza; e o carácter cooperativo da sociedade, que através da troca de mercadorias, troca se trabalho por trabalho.

O segundo autor, David Ricardo, tem em sua formulação o enfoque sobre a questão da distribuição - para Smith seria a riqueza - ou seja, o processo de distribuição da riqueza era uma dos seus principais questionamentos, conforme expõe Mattei (Mattei, 2003). E essa distribuição para Ricardo se dava entre três classes: o proprietário da terra, o dono do capital e os trabalhadores, conforme consta em sua obra Ricardo (Ricardo, 1996).

Ricardo, assim como Smith, trata do trabalho no início de sua Obra. O escritor londrino, traz como subtítulo de seu primeiro capítulo a definição sobre a Teoria do Valor do Trabalho:

“O valor de uma mercadoria, ou a quantidade de qualquer outra pela qual pode ser trocada, depende da quantidade relativa de trabalho necessário para sua produção, e não da maior ou menor remuneração que é paga por esse trabalho.” Ricardo (Ricardo, 1996) - *Princípio de Economia Política e Tributação* - Pg: 23

Dando continuidade à linha teórica, ele insere imputa a noção de utilidade para a mercadoria. Ou seja, para Ricardo, a utilidade é algo essencial para mercadoria na imputação de valor da mesma - mesmo considerando em alguns casos que o bem é escasso e nenhum trabalho poderá aumentar a oferta Ricardo (Ricardo, 1996). Mesmo assim, ressalta a importância tanto do valor do trabalho quanto na utilidade no processo de valoração, como percebe se neste trecho de sua obra:

“Possuindo utilidade, as mercadorias derivam seu valor de troca de duas fontes: de sua escassez e da quantidade de trabalho necessária para obtê-las.” Ricardo (Ricardo, 1996) - Princípio de Economia Política e Tributação - Pg: 24

Por fim, Mattei (Mattei, 2003), conclui que - mesmo com alguns pontos divergentes da teoria de Adam Smith - Ricardo reafirma que a quantidade de trabalho necessário para produzir as mercadorias determina, em parte, o valor desta. O trabalho nesta teoria é instrumental, e Ricardo se concentrou em parte para o estudo da distribuição do produto, entre as classes sociais colocadas anteriormente.

De uma maneira geral, percebe-se que os dois teóricos clássicos se preocupam em definir a Teoria do Valor do Trabalho com base no valor de uso das mercadorias. Ou seja, mesmo considerando que o trabalho é importante para a produção de mercadoria, ambos os autores têm como foco a utilidade da mesma. Fato este que Mattei (Mattei, 2003) aponta afirmando que a referida teoria sofre uma reformulação definitiva em Marx quando o mesmo desvenda que a mercadoria não se troca por trabalho, mas sim pela “capacidade de trabalho”.

2.1.2 Na Concepção Marxista

A Teoria do Valor do trabalho no pensamento de Karl Marx foi desenvolvida em suas obras: *O Capital* e no *Grundrisse*. Cabe ressaltar que em cronologia a obra *Grundrisse* é antecessora à obra *O Capital*, isso pode parecer algo simples, mas a relação de dependência entre as duas obras é clara. Muitos estudiosos afirmam que o *Grundrisse* seria o esboço do que foi *O Capital*, obra prima de Karl Marx. A forma de pensar desse escritor alemão, afirma, em linhas gerais, que o progresso das sociedades humanas se dá através da luta de classes entre os burgueses - controladores da produção, e o proletariado - os trabalhadores que fornecem sua mão de obra para produção.

É interessante perceber que certas passagens do *Grundrisse* são praticamente transcritas no *O Capital*, demarcando uma definição conceitual. Na verdade o *Grundrisse* marca o início da consolidação do pensamento econômico político de Marx, que terá fim na obra *O Capital*, escrita dez anos mais tarde. Sendo assim, buscar-se à intercalação dessas obras para melhor entendimento sobre o pensamento marxista e suas reflexões acerca da Teoria do Valor do Trabalho.

Voltando ao final do tópico anterior, temos que a visão clássica da Teoria do Valor do Trabalho se prende muitas vezes na relação entre trabalho e o valor de uso. Contudo, para Karl Marx, o conceito de mercadoria causa uma ruptura de pensamento com relação aos autores clássicos. Na obra *O Capital* ele assim define mercadoria no início do livro como sendo:

“A mercadoria é, antes de tudo, um objeto externo, uma coisa, a qual pelas suas propriedades satisfazem necessidades humanas de qualquer espécie.” Marx (Marx, 1996) - O Capital - Pg: 165

Para Marx, esse conceito é tão relevante que ele chega a afirmar que quem venha a produzir um objeto para uso próprio e direto, para consumo, está criando um produto e não uma mercadoria. O autor depois desta formulação - conceituação de mercadoria - complementa esse conceito, através da diferenciação entre valor de uso e valor de troca, que são dois fatores das mercadorias.

Inclusive começa essa distinção corroborando com o pensamento dos teóricos clássicos de que o valor de uso está relacionado com a utilidade de uma coisa. E que essa utilidade é determinada de acordo com as propriedades intrínsecas que formam a mercadoria (o ferro é diferente do ouro, que por sua vez é diferente da soja, por exemplo, mostram que estes produtos possuem qualidades diferentes). Marx acrescenta que, segundo Ricardo, o valor de uso de uma mercadoria tem se o pressuposto determinístico de quantidade. E por fim conclui que o valor de uso constitui conteúdo material da riqueza, - relembra Smith - independente de sua forma social.

Já o conceito de valor de troca para Marx se baseia na relação quantitativa num primeiro momento - na proporção da troca de valores de uso entre duas ou mais espécies Marx (Marx, 1996). Essa relação quantitativa de troca pode sofrer mutações constantes ao longo do tempo e espaço. Tal relação é o que diferencia Marx dos outros autores. Ele aponta o fato de que o valor de troca deve possuir um viés casual e relativo em sua formação. Marx frisa essa relatividade de valor da mercadoria que não se pode ser determinada sem que seja colocada uma mercadoria em relação com todas as outras. Esta relação de determinação de valor entre as mercadorias é peculiar e característica da sociedade burguesa.

O escritor alemão, ao final da discussão, conclui que os valores de uso das mercadorias são diferentes no carácter qualitativo, e os valores de troca difere no carácter quantitativo. Mattei (Mattei, 2003) diante desse contexto colocado por Marx, percebe a importância para a compreensão do processo de produção capitalista. Ele faz uma análise deste contexto proposto pelo autor afirmando que:

“(...)os valores de uso, enquanto produtos do trabalho de produtores privados, não são úteis socialmente, porque não satisfazem diferentes necessidades da sociedade de forma imediata. Isso só ocorre através da troca, momento em que esses valores de uso sofrem uma primeira metamorfose, pois despem-se de sua veste material e tornam-se, enquanto mercadorias, iguais a qualquer outro bem de uso, assumindo idêntica qualidade de valores de troca e diferindo uns dos outros apenas quantitativamente.” Mattei (Mattei, 2003) - Teoria do valor - trabalho: do ideário clássico aos postulados marxistas - Pg: 288

Toda essa fundamentação e discussão colocada fizeram com que o pensamento marxista colocasse de lado o valor de uso dos corpos das mercadorias, para perceber a propriedade de serem produtos do trabalho Marx (Marx, 1996). Ou seja, as mercadorias são produtos do trabalho humano, dispêndio de cérebro, nervos, mãos e sentidos do homem. Alerta logo depois desse apontamento que se abstrair do produto do trabalho o seu valor de uso, abstrai se os componentes e as formas corpóreas que fazem dele ter seu valor de uso. Deduz então o autor:

“(...) Ao desaparecer o carácter útil dos produtos do trabalho, desaparece o carácter útil dos trabalhos neles representados, e desaparecem também, portanto, as diferentes formas concretas desses trabalhos, que deixam de diferenciar-se um do outro para reduzir-se em sua totalidade a igual trabalho humano, a trabalho humano abstrato.” Marx (Marx, 1996) - *O Capital* - Pg: 168

Temos então que independente da forma de um objeto, o trabalho humano despendido no processo de produção desse produto, gera uma cristalização dessa substância social que é a mercadoria ter seu valor mercantil. Ou seja, o trabalho cria um padrão qualitativo das

mercadorias, permitindo uma formação no valor de troca referente ao aspecto quantitativo da mercadoria.

Daí percebe-se outra diferença de Marx para com os outros autores, pois o escritor alemão, enfatiza a importância do trabalho no processo de produção de uma mercadoria. Enquanto muitas vezes para os teóricos clássicos, o trabalho seria algo imediato e muitas vezes indiferente às mercadorias. Já Marx, considera que na mercadoria há o resultado de um trabalho humano, e isso gera valor para a mesma. O trabalho não é valor, mas, é um dos fundamentos Mattei (Mattei, 2003).

Outro ponto é o carácter social do trabalho na teoria marxista. A mercadoria deve estar ligada a uma necessidade social. E o trabalho incorporado nesta deve conforme relata Marx (Marx, 1996), representar uma parte da soma global de trabalho realizado pela sociedade. Ou seja, a capacidade de realizar trabalho é uma necessidade humana e deve ser socializada. Para que o homem se objetive no mundo, é necessário que ele o faça por meio de uma atividade produtiva. Essa atividade o humaniza na mesma medida, e da mesma forma que humaniza o mundo no qual ele vive.

Analisando todos os pontos já colocados, é possível ter uma maior clarividência sobre a Teoria do Valor do Trabalho para Marx. O autor desenvolve sua teoria afirmando que:

“(...) O valor de uma mercadoria é determinado pela quantidade de trabalho incorporado ou cristalizado nela, queremos referir-nos à quantidade de trabalho necessário para produzir essa mercadoria num dado estado social e sob determinadas condições sociais médias de produção, com uma dada intensidade social média e com uma destreza média no trabalho que se emprega.” Marx (Marx, 1996) - *O Capital* - Pg: 94

Na sua obra anterior - *Grundrisse* -, o autor ratifica a teoria colocada, afirmando que o pressuposto do valor da mercadoria é e continua sendo a massa do tempo de trabalho imediato. Afirmando que o quantum de trabalho empregado é um fator decisivo na produção de riqueza Marx (Marx, 2011). A quantidade de trabalho social que é empreendido nas mercadorias, determina o valor de troca como já colocado, se cresce a quantidade de trabalho exigida para a produção da mercadoria, tem-se o aumento no seu valor Marx (Marx, 1996). Ou seja, o tempo de trabalho aplicado implica diretamente no valor das mercadorias.

E na esfera social que é um dos diferenciais na formulação teórica de Marx, sustenta que os valores de troca das mercadorias são funções sociais. As mercadorias possuem objetividade de valor que são uma expressão social quando observadas as relações de troca entre elas. Uma unidade social do trabalho humano, onde cada trabalhador contribui com sua mercadoria nessa troca satisfazendo uma necessidade social qualquer. Tal trabalho é criador de valor socialmente necessário, quando executado em condições médias vigentes de técnica, destreza e atenção do operário e a questão do esforço e tempo despendido na realização da tarefa produtiva.

2.2 Sociedade da Informação

A Revolução Informacional começa na segunda metade do século XX, entre as décadas de 60 e 70. Diante de uma crise no modelo produtivo dominante, da Revolução Industrial, houve uma renovação nos processos produtivos, principalmente neste contexto

pós Segunda Guerra Mundial. A automação das esteiras rolantes e controles eletrônicos, são exemplos de uma nova forma de ampliação do trabalho do operário. Temos na mesma época o começo da fabricação dos computadores agora em nível industrial, que outrora foram usados durante a segunda grande guerra. Ou seja, temos um conjunto de transformações no processo de produção de mercadorias, tendo o avanço tecnológico como protagonista. Muitos autores afirmam que estas mudanças gerou uma nova forma de organização industrial, e da relação entre capital e trabalho.

“O Computador é um instrumento de transformação do mundo material e humano. O computador é também uma condição material essencial para elevação da produtividade do trabalho em todas as esferas de atividade.” Lojkine (Lojkine, 1995) - A Revolução Informacional - Pg: 49 e 50

A Sociedade da Informação tem na Revolução Informacional sua base, sendo esta num primeiro momento, uma revolução tecnológica de conjunto, que se segue após à Revolução Industrial, Lojkine (Lojkine, 1995). No cenário que se apresenta a Sociedade Informacional podem se destacar três pontos que são característicos desta nova Sociedade:

- Produção e Informação: além da produção a informação tem papel fundamental nesta nova sociedade que veio através da revolução tecnológica;
- Assalariados da Produção Material x Assalariados encarregados de “Tratar a Informação”: nesta nova sociedade cresce o número de profissionais que manipulam e tratam a informação. O profissional agora atravessa um processo de reprofissionalização que atinge tanto o setor público como o privado, principalmente os assalariados do setor de Serviços;
- Mudança do Trabalho Humano de Manipulação para o Tratamento da Informação: a Revolução Informacional nasce da dicotomia entre a revolução da máquina ferramenta (caracterizada pelas funções manuais) e a revolução da automação (caracterizada pelo uso de certas funções cerebrais desenvolvidas pelo maquinismo industrial). Ou seja, a imagem do Músculo - símbolo da Sociedade Industrial; é substituído pela imagem do Cérebro - símbolo da Sociedade Informacional, quem simboliza o tratamento da informação;

Outra característica que pode ser apontada é o Instrumento Informático que pode permitir conectado a outras novas técnicas de telecomunicação, a criação, circulação e a estocagem de uma imensa massa de informações. A estrutura Organizacional continua sendo hierárquica, ou seja, não há mudança na divisão social do trabalho, ou seja, os confrontos de poder que há na base das grandes organizações, continuam. Contudo o instrumento informático e o tratamento da informação trazem uma nova estrutura tripartida, segundo Lojkine (Lojkine, 1995) :

- Os que decidem de modo não programado - Dirigentes Inovadores;
- Os que decidem sobre o funcionamento cotidiano da organização - Executivos Operacionais;
- Os que se encarregam dos “processos de base do trabalho” - Operadores;

Mesmo tendo uma Estrutura Organizacional baseada na Sociedade Industrial, passa a existir uma mudança importante, pois devido à invenção científica moderna, o seu desenvolvimento se dá mediante a um trabalho em equipe, através da cooperação (em dicotomia ao modelo antigo que envolvia a troca de mercadorias entre proprietários privados - Cooperação x Troca). A informática tem discutido, as antigas divisões das funções da empresa e aprimoramento de outras - como o departamento de estudo aos serviços pós venda, muito utilizado em empresas físicas e virtuais (e-commerce). Outro ponto são as novas relações entre empresa que empreita e subempreiteiras, empresas industriais e empresas de serviços - tais como laboratórios de pesquisa ou exames médicos, empresa de banco de dados, provedores de servidores web, entre outros. Para uma melhor ilustração deste cenário temos a figura 2.1.

As pirâmide de poderes em face dos diferentes tipos de polivalência

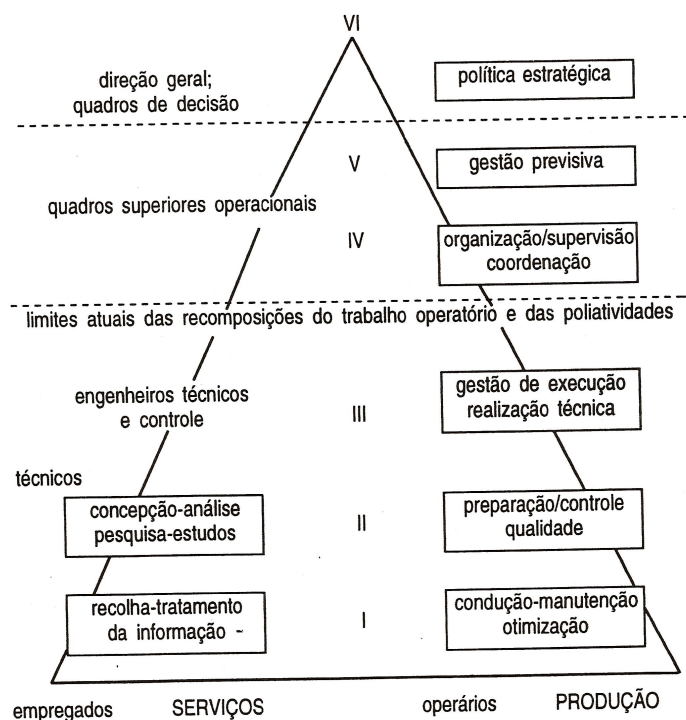


Figura 2.1: Pirâmide de Poderes da Sociedade Informacional - Lojkin (Lojkin, 1995).

Desta figura ressalta-se que a partir do nível IV temos as funções dos quadros superiores operacionais, sendo o nível VI o nível estratégico. A crítica nesta disposição da sociedade informacional permanece a mesma da sociedade industrial, onde existem os que pensam e os que apertam parafusos. Daí percebe-se que a estrutura de poder e os critérios de gestão não mudaram significativamente:

- Nível I: responsável pela condução, controle de qualidade e otimização;

- Nível II: uma “extensão do primeiro nível”, abrangendo a preparação, programação e controle de qualidade;
- Nível III: comando imediato, responsável pelo lançamento, acompanhamento e o controle dos procedimentos, que seria a função dos antigos supervisores;
- Nível IV: Exame de regras e dos procedimentos, determinação de recursos e de tarefas;
- Nível V: responsável pela preparação do futuro, definição de objetivos, concepção organizacional, avaliação final;
- Nível VI: as atividades de definição de regras, normas critérios de gestão, decisão nas grandes operações financeiras e de investimento;

Vejamos a diferenciação que Lojkine (Lojkine, 1995) faz entre as características da Revolução Industrial e da Revolução Informacional, de maneira a ilustrar suas oposições, mas também uma certa evolução de uma para outra:

- Revolução Industrial:
 - A Especialização (com a fundamental oposição trabalho manual trabalho intelectual), as máquinas-ferramenta especializadas;
 - A Estandarização de peças, pautas e produtos. Mas também das tarefas humanas;
 - A Reprodução Rígida (continuidade da cadeia) - transmissões mecânicas, cadeias e sequências rígidas no processo produtivo;
- Revolução Informacional:
 - A Verdadeira Polivalência ou Poli funcionalidade, relacionada tanto quanto à organização dos meios de trabalho, quanto à organização dos homens.
 - A Flexibilidade, a variedade dos usos possíveis das máquinas informacionais. Tem se a auto regulação dos sistemas flexíveis na automação;
 - A Estrutura em Redes Descentralizadas;

Pode se notar que os meios informáticos de trabalho trazem um novo contexto mundial relacionado a objetivação pela máquina, de funções abstratas, reflexivas do cérebro. Esta revolução tecnológica traz um avanço dos limites da revolução industrial, através de um processo que começou nela mesma, que é o processo de automação. Pode se observar mais claramente outros pontos desta diferenciação através da tabela 2.1

O processo de automação tem como fim o controle da produção, e com o desenvolvimento da eletrônica e informática, colaborou para uma mudança da função que antes era de manipulação nas máquinas, para a função sensível reflexiva na monitoração dos processos automatizados. Cada vez mais a força física do modelo Industrial vem sendo substituída pela força mental que controla e dirige a “mão inteligente” das máquinas.

Revolução Industrial	Revolução Informacional
- Máquina operatriz e máquina ferramenta	- Máquina auto regulada
- Sistema de máquinas automáticas especializadas	- Sistema flexível, autorregulado, de máquinas
- Continuidade, rigidez, standardização	- Polifuncionais
- Objetivação da mão de obra	- Objetivação de funções cerebrais abstratas (direção, regulação da máquina)
- Objeto de trabalho: ferro e carvão	- Silício, supercondutores
- Meios de circulação materiais centralizados e segregados	- Meios de circulação materiais e imateriais (informacionais) descentralizados e interativos (telemática em rede)
- Oposição ciência/produção	- Inter-relações ciência / produção
- Domínio das atividades industriais	- Cooperações serviços- indústrias
- Papel dinâmico da indústria metalúrgica	- Papel dinâmico da mecatrônica
- Segregação urbana	- Integração urbana de funções em rede

Tabela 2.1: Confronto entre Revolução Industrial e Revolução Informacional - Lojkine (Lojkine, 1995). Com adaptações

2.3 Trabalho Imaterial

A teoria do valor do trabalho muitas vezes é questionada no atual cenário mundial. Não obstante, o que se percebe são estudiosos que buscam hoje, uma releitura desta teoria. Antunes (Antunes, 2008) um dos grandes estudiosos brasileiros debruçado sobre essa discussão afirma em uma de seus vários artigos sobre o assunto:

“O trabalho imaterial expressa a vigência da esfera informacional da forma mercadoria: ele é expressão do conteúdo informacional da mercadoria, exprimindo as mutações do trabalho no interior das grandes empresas e do setor de serviços, onde o trabalho manual direto está sendo substituído pelo trabalho dotado de maior dimensão intelectual.” - *Desenhando a nova morfologia do trabalho: As múltiplas formas de degradação do trabalho* - Pg: 26

Com isso percebe-se o atual cenário entre a subjetividade do trabalhador e a máquina inteligente que necessita desta interação. É a crescente figura do trabalho imaterial verificada e constatada como um dos novos cenários de atuação do trabalhador. Contudo, Antunes (Antunes, 2008), considera que algumas formas de trabalho imaterial são expressão, muitas vezes, de modalidades de trabalho vivo, que não deixam de ser uma valorização do trabalho humano. Fato que Marx já prefigurava em sua obra quando afirma:

“Mas há atividades que não produzem bens materiais e, contudo, são necessárias ao processo de produção ou o prolongam na esfera da circulação, devendo ser consideradas produtivas e, portanto, criadoras de valor e mais-valia.” Marx (Marx, 1996) - O Capital - Pg: 39

É neste ponto que Rosso (Rosso, 2014) coloca a necessidade de verificação caso por caso para verificar se o trabalho está produzindo uma mercadoria, valor e mais valia. Em sua análise sobre a teoria do valor do trabalho, ressalta que para Marx, a materialidade contraposta a razão ou o espírito. Com isso, tem-se que para o filósofo alemão, a visão de mundo tem seu fundamento nas relações dos seres vivos em ação como discutido anteriormente. Daí tem-se duas correntes filosóficas que desenvolvem e discutem a imaterialidade do trabalho no setor de serviços. A primeira analisa a composição setorial da força de trabalho, ou seja, consiste na divisão da força de trabalho em três setores de atividade econômica, primário (agricultura), secundário (indústria) e terciário (serviços). A partir dessas denominações pode-se observar dois modelos de transformações. Nos países capitalistas e ricos, ao passar do tempo, a população ocupava cada um desses setores de acordo com as transformações que os países viviam, passando de uma fase majoritariamente na agricultura para a indústria e depois o setor de serviços. De forma distas, nos países que entraram tardiamente no sistema capitalista ou periféricos, a fase de ocupação saltou de prioritariamente primária para o setor de serviços. Segundo o autor, muitos desses postos de trabalhos possuem baixa qualidade apesar de terem uma grande força de trabalho.

A segunda corrente filosófica, estuda os impactos da chamada revolução da informática sobre a estrutura de produção do setor de serviços e questiona os “supostos limites inquestionáveis da teoria do valor do trabalho”, Rosso (Rosso, 2014). Ele revela que segundo a ótica desta corrente, representada pela produção filosófica sociológica, a teoria deixaria de aplicável quando os trabalhadores deixassem de produzir valor diretamente, ou seja, quando o trabalho humano não fosse mais necessário para produzir mercadoria. O autor ainda enumera algumas fragilidades encontradas nesta segunda vertente. As três que se destacam é, primeiramente ele afirma que toda a mercadoria exige o emprego de capacidades físicas, mentais e afetivas de quem a produz, mesmo que se converta em bem material. Outro fator que Rosso destaca é que, diferente do que alguns pensadores afirmam, o tempo médio socialmente necessário ainda é utilizado para cálculo de remuneração de salários no mercado capitalista. Ele também afirma ser importante esse mecanismo para prevenir jornadas de trabalho excessivas. Por fim, ele destaca que diferente do que os autores da crítica afirmam, o trabalho imaterial e material tem a mesma natureza e em todos os tipos de atividade profissional o trabalhador investe energias físicas, intelectuais e afetivas.

Para o autor, Marx acreditava que não é só o trabalho material que é produtivo, mas quando o trabalho imaterial gera diretamente mais-valia ele também se transforma em produtivo. Dessa forma, trazendo a lógica para a atualidade, o autor argumenta que:

“A mesma atividade profissional pode ser produtiva se organizada sob a forma de uma exploração do trabalho vivo por indivíduo ou empresa, ou improdutiva se significa simplesmente a troca de serviços por dinheiro, como acontece com muitos profissionais liberais e científicos que operam individualmente e são auto empregados.” Rosso (Rosso, 2014) - Teoria do valor e trabalho produtivo no setor de serviços - Pg: 82

Para ratificar a linha de raciocínio, o autor afirma que Marx também entendia que o trabalho imaterial poderia ser produtivo desde que atendesse o critério de produzir mais-valia, como já relatado acima. Rosso (Rosso, 2014) faz uma crítica a ideia reducionista muito disseminada atualmente, que surgiu da obra de Marx, em que só se considera trabalho produtivo aquele que resulta um produto material. Sobre este pensamento o

autor adverte que o importante é analisar se a relação vem a ser de troca simples de serviço ou objeto por dinheiro, ou se o trabalho visa aumentar o valor do capital.

Para o referido autor o valor pode ser entendido como uma relação social entre pessoas que venha a assumir uma forma material e está relacionado ao processo de produção. Dessa forma chega se a seguinte definição:

“Serviços produzem mercadorias, valor e mais valia quando estiverem presentes as condições de serem executados por trabalho assalariado contratado, com o fim de realizar ganhos e se, por meio do trabalho, houver produção e circulação de mercadoria de tal modo que o valor se realize, resultando em mais valia que pode ser apropriada pelos contratadores ou por outras agências do capital.” Rosso (Rosso, 2014) - Teoria do valor e trabalho produtivo no setor de serviços - Pg: 85

Nesta conjuntura Antunes e Alves (Antunes & Alves, 2004), retomam o cenário de interação entre a subjetividade de quem trabalha e o novo maquinário inteligente. E dessa relação que almeja a criação de valor e mais valia, num curto espaço de tempo, e num contexto onde há uma gama de informações sendo produzidas, podendo a partir deste envolvimento interativo entre homem e máquina, gerar um estranhamento do trabalho, ou até mesmo, uma alienação. A crítica na literatura encontra se neste fato, na tentativa da busca de uma solução para a dicotomia entre trabalho produzido e o valor do trabalho operado nesta produção. E o presente trabalho pretende atacar esse problema na produção de software, assim como em outros setores produtivos. Sendo assim, para melhor elucidação sobre o tema, vejamos alguns pontos relativos ao setor de serviços.

2.4 Setor de Serviços

O ramo de serviço é um dos que mais empregam pessoas, e tem ganhado destaque no contexto mundial. O setor corresponde às atividades de comércio de bens e prestação de serviços, abrangendo várias atividades relacionadas a computação incluindo desenvolvimento e produção de softwares, recarga de cartuchos de impressora, manutenção de computadores e notebooks, entre outros. Segundo dados da ABES (ABES, 2014) Associação Brasileira das Empresas de Software, com relação a distribuição de TI à nível mundial, o ramo de serviços tem a parcela de 31% de mercado comparado com outros ramos da computação como hardware e software, como percebe se pela Figura 2.2.



Figura 2.2: Distribuição do Mercado de TI à nível Mundial - (ABES, 2014).

À nível de Brasil, continuando este estudo, o mercado doméstico de Tecnologia da Informação, que engloba às áreas de: hardware, software e serviços, movimentou 60 bilhões de dólares em 2014, representando 2,6% do PIB brasileiro e 3% do total de investimentos de TI no mundo. O setor de serviços apresentou um crescimento de 7,3% sobre o ano

anterior, movimentando cerca de quase 20 milhões de dólares (Figura 2.3) no mercado doméstico, ou seja, sem contar as exportações destes serviços.

Mercado Total de TIC no Brasil - 2014 (US\$ Milhões) / Total ITC Market in Brazil - 2014 (US\$ Million)				
	Segmentação Mercado Market Segmentation	Mercado Doméstico Domestic Market	Mercado Exportação Export Market	Mercado Total Total Market
TI / IT	Software / Software	11.215	225	11.440
	Serviços / Services	13.967	633	14.600
	Hardware / Hardware	34.841	326	35.167
	Total TI / IT Total	60.023	1.184	61.207
TIC / ITC	Telecom / Telecom	98.027	-	98.027
	Total TIC / ITC Total	158.050	1.184	159.234
TIC + Outros / ITC + Other	Outros Serviços / Other Services (*) (**)	5.096	2.218	7.314
	TI "In House" / "In House" IT (**)	48.255	-	48.255
	Total Mercado TIC / ITC Total Market	211.401	3.402	214.803

Figura 2.3: Mercado de TI no Brasil - (ABES, 2014).

Cerca de 86% dos softwares de serviços são desenvolvidos dentro do Brasil, isto é, o mercado de serviços no Brasil se encontra em grande ebulição (Figura 2.4) e movimentando grandes quantias de valores. Em grande parte com a população tendo mais acesso à computadores, *smartphones*, *tablets* favorecem o crescimento de serviços e locais de assistência técnica, compra e venda de peças e acessórios, dentre outros, que venham a atender as demandas desse mercado que vem crescendo ao longo do tempo. Outro fato que venha a justificar esse crescimento reflete no gradativo processo de informatização das empresas que solicitam diversas vezes programas para sua automação, emissão de notas fiscais, compra e assistência de produtos. Dessa forma percebe que a sociedade brasileira de forma geral tem vivido esse processo de informatização em vários setores da sociedade tanto público, quando privado.

Foram identificadas cerca de 12.660 empresas dedicadas ao desenvolvimento, produção, distribuição de software e de prestação de serviços no mercado nacional, sendo que 55% delas com atividade principal voltada para o desenvolvimento e produção de software ou prestação de serviços. Essas empresas vão desde grandes empresas multinacionais a micros e pequenas empresas. Um último dado de grande importância neste estudo, está na amostra de indicadores de evolução do mercado de softwares e serviços, num período de dez anos (Figura 2.5). Com os dados apresentados, percebe se a evolução do mercado de serviços, já tendo sido discutido anteriormente. Esses números revelam que os donos de empresas do ramo de serviços estão diante de um cenário de franca expansão do mercado, movimentando maiores quantias, podendo agregar um aumento do lucro da empresa. Contudo, a figura do trabalhador que contribui para esse crescimento e desenvolvimento desse mercado, por diversas vezes é tratada como secundária, nem dada tanta ênfase quanto aos números aqui apresentados.

Uma das dificuldades com relação ao trabalho neste ramo de serviços está no fato de não se ter modelos ou métricas que trabalhem a ideia do esforço do trabalho. Daí

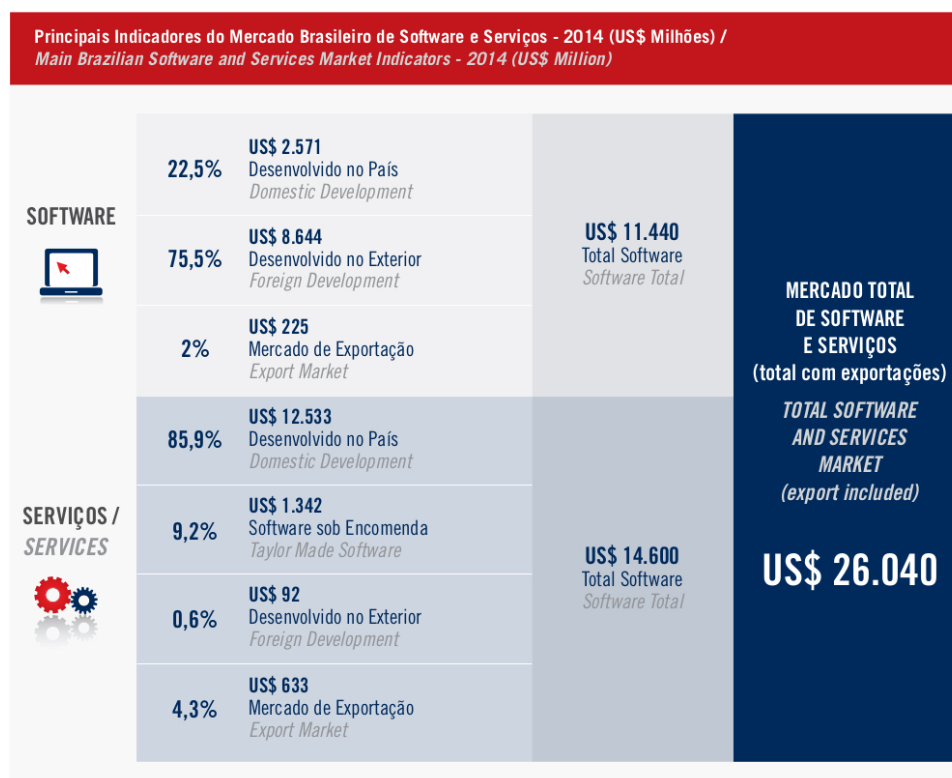


Figura 2.4: Panorama de Serviços no Brasil - (ABES, 2014).

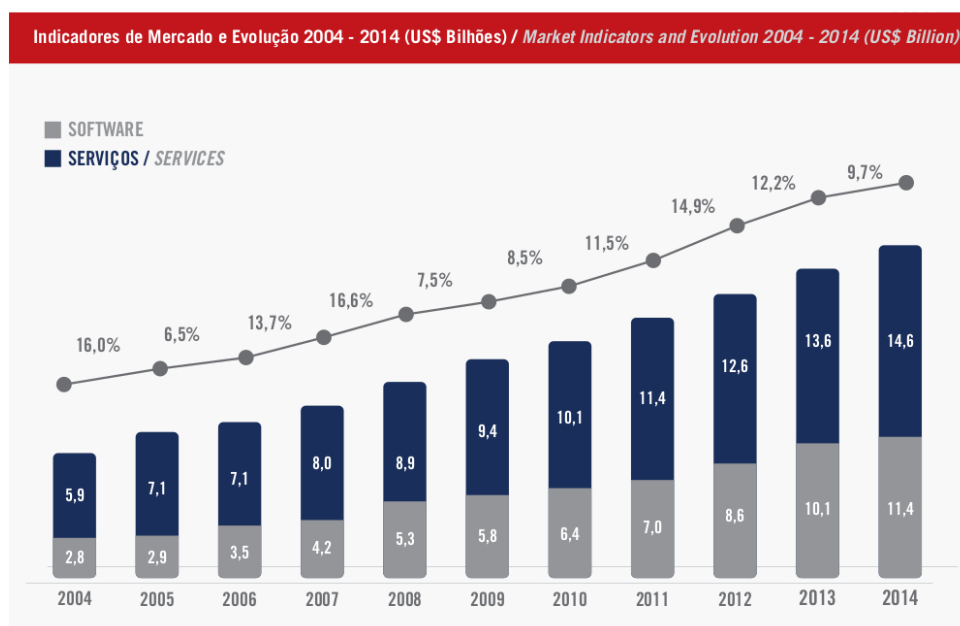


Figura 2.5: Indicadores de Mercado e Evolução - (ABES, 2014).

tem a crítica marxista entre valor material e imaterial: o valor do trabalho material tem como resultado algo tangível, palpável, a grosso modo; e o valor imaterial como algo intangível, não se toca num software, por exemplo. Autores no ramo da sociologia apontam que o fato da imaterialidade deste tipo de trabalho no ramo de serviços é um fator importante a ser considerado nesta nova sociedade informacional, tendo em vista que na sociedade industrial o trabalhador produzia um quantitativo de peças num dia e esta seria uma forma de mensurar o trabalho realizado naquele período. Antunes e Alves (Antunes & Alves, 2004) traz que as mutações organizacionais, tecnológicas e de gestão também afetaram fortemente o mundo do trabalho nos serviços, que cada vez mais se submetem à racionalidade do capital e à lógica dos mercados. Várias atividades no setor de serviços anteriormente consideradas improdutivas tornaram-se diretamente produtivas, subordinadas à lógica exclusiva da racionalidade econômica e da valorização do capital. Há então uma dificuldade atual de mensurar o esforço do trabalho neste novo paradigma da sociedade informacional, no trabalho imaterial, como a produção de softwares.

Capítulo 3

O Desenvolvimento de Software

O presente capítulo traz a apresentação da área de engenharia de software. Principalmente os assuntos que sustentarão a elaboração do novo modelo de mensuração do esforço do trabalho proposto no próximo capítulo. Buscando através dos tópicos deste o básico dos assuntos para apoio à proposição do novo modelo.

A engenharia de software é uma disciplina nova que visa o desenvolvimento de sistemas de software com boa relação entre custo e benefício. Entende-se por software não apenas um programa de computador em si, mas toda a sua documentação associada e dados de configuração para a operacionalização dos mesmos. O software é abstrato e intangível, não possuindo restrições num primeiro momento, nem regido por processos industriais ou leis físicas. Porém essa ausência de limitações o tornam complexo, e muitas vezes com dificuldade de compreensão Somerville (Somerville, 2007).

Processos técnicos de desenvolvimento, gerenciamento de projetos, desenvolvimento de ferramentas, métodos e teorias são algumas das atividades executadas por esta engenharia. Daí percebe-se a importância da mesma no processo de produção de softwares.

3.1 Processo de Produção de Software

O processo de software gera através de atividades e resultados associados, um software, conforme a finalidade e uso do mesmo. Dentro da literatura temos uma gama de processos de produção de software, que se resume no conceito anteriormente apresentado. Contudo cabe ressaltar que estes processos são intelectuais, e num primeiro momento intangíveis, imateriais, porém ao final de todo processo tem-se um produto (o software) final resultante. Para Pressman (Pressman, 2010), o processo de software é um arcabouço para as tarefas necessárias na construção de softwares, sendo no processo a definição da abordagem adotada na elaboração dos softwares, e o estabelecimento do contexto onde os métodos técnicos serão aplicados.

Produto que depende do julgamento humano para verificação se está realmente conforme solicitado ou não. Tendo esta diversidade de processos podemos extrair algumas atividades fundamentais que são realizadas. Estas atividades básicas se dividem em etapas, tomando por base as apresentadas por Somerville (Somerville, 2007):

1. Proposição do Software: onde é passado qual produto que deve vir a ser elaborado e qual será a destinação, ou seja, o propósito/finalidade do software;

2. Levantamento de Requisitos: nesta etapa verifica se junto ao cliente, quais os requisitos básicos que o software necessita. Por exemplo: quais campos serão usados, como será feita a distinção entre estes campos, como será feita a ligação entre eles, revisão das variáveis usadas;
3. Análise de Requisitos e Desenvolvimento do Software: nesta fase é feito um estudo e análise dos requisitos levantados para elucidar o que o sistema fará, para daí então pensar como o sistema será feito. A partir desta diferenciação é feita a divisão do programa, na maioria das vezes, em pequenos módulos para um melhor compartilhamento do trabalho, redução do tempo de confecção do mesmo, redução de custos para manutenção, entre outros;
4. Validação, Verificação e Testes: revisão dos módulos do programa validando às necessidades do cliente e verificando se os requisitos passados estão sendo atendidos. Os testes são feitos para atender essa validação e verificação, onde pode ser encontrado erros e o comportamento do programa em vários aspectos, sempre visando a garantia que ele faça o que o cliente solicitou na primeira fase;
5. Apresentação ao Cliente e Evolução: exposição do software ao cliente que verificará se o produto está conforme pedido; implementação no ambiente do cliente, e evolução do sistema para atender alguma necessidade mutável do mesmo.

É importante ressaltar que não há o melhor processo de software, pois cada projeto de construção de software pode haver mudança no processo de produção, ou então uma releitura e aprimoramento de um processo que foi usado como base. Com isso, os termos qualidade e confiabilidade têm merecido uma atenção especial dos pesquisadores quando se foca o processo de desenvolvimento de software moderno. E para garantir essa qualidade e confiabilidade, Pressman (Pressman, 2010) aponta algumas atividades de apoio complementando que podem ser aplicadas ao longo do processo de produção de software, entre elas: acompanhamento e controle de projeto de software, que avalia o progresso do software com base no plano de projeto, tomando as ações necessárias para a manutenção do programa; garantia de qualidade de software, trabalha a definição e condução das atividades para o garantir a qualidade; revisões técnicas formais, buscam descobrir e remover os erros antes que sejam propagados para outra atividade na elaboração do software; medição, busca a definição e uso de medidas de processo, projeto e produto que venham a auxiliar na entrega do software, satisfazendo a necessidade do cliente.

Os processos usados para desenvolver um projeto de software têm a maior importância na qualidade do software produzido e na produtividade alcançada pelo projeto. Por consequência, existe uma necessidade de melhorar os processos usados em uma organização para desenvolver os projetos. Ter um amplo conhecimento sobre processos de software é fundamental quando se venha a adaptar um processo de software maduro para cada produto solicitado e conforme as demandas do mercado Pressman (Pressman, 2010). Sendo assim a importância de um processo de software pode já ajudar a uma padronização na elaboração e construção de software de uma empresa/organização. Ressaltando que dentre os benefícios desta padronização, podemos apontar:

- Redução no tempo e custo de elaboração de outros softwares, tanto para os treinamentos daqueles que trabalharão em algum projeto da empresa;

- Apoio à introdução de novos processos que futuramente possam vir a ser incorporados;
- Reduzir o número de defeitos nos softwares que venham a ser produzidos e distribuídos pela empresa;

Cabe ressaltar que estas melhorias no processo devem sempre ser uma atividade complementar a uma empresa. Para que de certa forma, não paralise os processos já usados, prejudicando novas solicitações de softwares, ou manutenção dos já produzidos. O sucesso na melhoria depende de um comprometimento da organização, planejamento e orçamento associados. Outro ponto a se ressaltar é a importância da comunicação com o cliente, os *feedbacks* são fundamentais durante o processo tanto no planejamento do desenvolvimento, quanto a avaliação dos custos e da viabilidade do software.

Para bem ilustrar este cenário, The Standish Group (Group, 2015) - que desenvolve e fornece ao mercado soluções avançadas que podem reduzir o risco e acelerar a agregação de valor dos investimentos em tecnologia da Informação - emite relatórios sobre boas práticas para o desenvolvimento e gerenciamento de softwares; informações relevantes sobre como anda o mercado de produção de softwares. Esse relatório é chamado de CHAOS Knowledge Center (CKC), e tem se alguns dados retirados deste relatório no ano de 2010 Group (Group, 2010) :

- 16,2% dos projetos são concluídos dentro do prazo e do orçamento. E todas as características e funções mantidas conforme a especificação inicial;
- 52,7% estão concluídos e operando. Doravante, acima do orçamento e extravasado a estima de tempo para sua conclusão. Neste caso, o software oferece menos características e funções do que o originalmente especificado;
- 31,1% dos projetos são cancelados em algum ponto durante o ciclo de desenvolvimento / processo de produção do software;

Tais dados evidenciam a importância do processo de produção de software e seus modelos, ou seja, todo o processo de desenvolvimento e criação de um software devem primar pela garantia de um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado com o cliente.

3.2 Modelo de Processo de Software

Tendo o entendimento das fases e da importância do processo de software, cabe agora falar dos modelos que abrangem o processo. Tem se como definição que os modelos de processo são uma representação do processo de uma maneira abstrata e simplificada, apresentando uma visão geral deste. Assim como no processo o modelo de software deve ser adaptado ao problema proposto, ao projeto que está sendo criado, à equipe que participará do projeto e à cultura organizacional que a empresa venha a ter como atesta Pressman (Pressman, 2010). Muitos consideram estes modelos como *frameworks* de processos, ou seja, conjuntos de classes usados cada qual para um problema semelhante - exemplo: processos de um tipo x de software usam se o modelo X', já processos do tipo

y com o modelo Y'. Contudo não é possível ver os detalhes de cada atividade específica, os modelos que serão apresentados são genéricos tratando a perspectiva arquitetural.

Somerville (Somerville, 2007), apresenta três modelos gerais: *workflow*, que mostra a sequência de atividades ao longo do processo, as atividades são representadas pela ação humana; fluxo de dados, considera processo como um conjunto de atividades, cada atividade pode realizar alguma transformação de dados. As atividades representam transformações realizadas por pessoas ou computadores; papel/ação, traz os papéis das pessoas envolvidas no processo e as atividades que cada um exerce. Destes modelos gerais, também chamados de paradigmas de desenvolvimento, advém os modelos de processos mais usados.

Pressman (Pressman, 2010), atesta que os modelos de processo diferem entre alguns pontos, entre esses podemos destacar:

- Fluxo geral de atividades e tarefas, e suas interdependências;
- Grau que as tarefas de trabalho são definidas em cada atividade;
- Grau de detalhes e rigor na descrição dos processos;
- Grau de envolvimento de clientes e outros interessados;
- Grau de prescrição da organização da equipe e seus papéis;

Atualmente na literatura, encontra-se uma distinção entre dois tipos de modelo: os modelos prescritivos de processo e os modelos de processo ágil. Cada um contém vários modelos, contudo nem todos serão contemplados neste estudo.

Pressman (Pressman, 2010), define que os modelos prescritivos de processo focam a definição, identificação e aplicação detalhada tanto das atividades como das tarefas de processo. Estes modelos são os que vem sendo usados ao longo dos últimos trinta anos. Tem-se como objetivo melhorar a qualidade do software tornando-os mais gerenciáveis, para guiar as equipes de trabalho as datas de entrega e os custos são mais previsíveis. Porém em algumas circunstâncias não se alcança os objetivos, e os modelos podem tornar-se mais burocráticos gerando uma maior dificuldade na construção dos sistemas. Já os modelos de processo ágil focam a agilidade no processo, tem como base princípios que dão num primeiro momento uma abordagem informal, e objetivam a manobrabilidade e a adaptabilidade, que podem ser vistos muitas vezes na criação de sistemas *Web*. Neste trabalho serão abordados sob a ótica prescritiva os modelos: cascata, incremental e espiral, e no caso da metodologia ágil o RUP (*Rational Unified Process*). Mesmo divergindo em alguns pontos ambos os tipos de modelo de processo buscam criar softwares de qualidade e que consigam satisfazer às necessidades dos clientes.

3.2.1 Modelo Cascata

O modelo cascata é um dos mais antigos modelos, é utilizado principalmente quando os requisitos de um determinado problema são bem compreendidos e a forma de trabalho venha a ser linear ao longo do processo. Uma forma de utilizar o modelo cascata é quando precisamos fazer adaptações ou aperfeiçoamentos em um sistema já existente: quando temos um sistema já pronto e precisamos fazer uma adaptação por conta de uma nova normal, ou mudança pontual no software. Os requisitos devem ser bem definidos e estáveis.

Este modelo, por vezes também chamado de ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, segundo Pressman (Pressman, 2010). Há um encadeamento de uma fase com outra, e apresentaremos estas fases relacionando com os autores **Somerville (Somerville, 2007)** e *Pressman (Pressman, 2010)*, respectivamente:

1. **Análise e definição de requisitos** ou *Comunicação*: início do projeto e especificação do sistema, ou seja, tem-se a definição dos objetivos, serviços e restrições do software. Tais informações por meio de uma consulta / conversa com o usuário;
2. **Projeto de sistema e software** ou *Planejamento*: divisão dos requisitos em hardware e software. Estabelecimento da arquitetura geral dos sistemas, onde começam as estimativas de tempo através da definição de um cronograma;
3. **Implementação e teste de unidade** ou *Modelagem*: realização do projeto do software através de um conjunto de programas, verifica-se cada unidade do software atende a sua respectiva especificação;
4. **Integração e teste de sistema** ou *Construção*: cada unidade do programa, elaborada e testada na fase anterior, são integradas, codificadas e testadas completando todo o sistema e verificando se os requisitos foram atendidos;
5. **Operação e manutenção** ou *Implantação*: é feita a entrega do software ao cliente e onde é feita a instalação e o mesmo é colocado em operação. Nesta fase ocorre a manutenção de pequenos erros, antes não detectados, ou o aprimoramento de novas funcionalidades. Por isso é uma das fases mais longas do ciclo de vida do processo;

De forma resumida, como cada autor aborda o modelo cascata nas figuras: 3.1 e 3.2

Neste modelo, em cada passo tem-se uma documentação registrada e a fase seguinte não pode passar para fase posterior sem seu término. Diante disso, o presente modelo tem hoje questionamentos sobre sua eficácia, devido há alguns problemas como: dificuldade em seguir o fluxo sequencial proposto pelo modelo, dificuldade em o cliente ter clareza sobre todos os requisitos do sistema e suas necessidades, e a demora depois de alguns passos para entregar uma versão ao cliente/usuário. E por fim há o bloqueio de alguns membros da equipe que precisam esperar que alguns completem as tarefas para que sejam dadas a sequência do trabalho, podendo interferir o tempo gasto na elaboração do projeto até sua conclusão. Não é um modelo que se enquadre num cenário de mudanças e ambientes imprevisíveis.

3.2.2 Modelo Incremental

Alguns projetos de software definem requisitos iniciais de software razoavelmente bem definidos, sendo assim, pode ser necessário o rápido fornecimento de um determinado conjunto funcional aos usuários. Para que os usuários vejam o que está sendo feito e que após esse fornecimento, tenha a possibilidade melhorar e expandir suas funcionalidades em versões de software posteriores. Neste cenário de possíveis e prováveis mudanças ao longo da produção do software, vê-se a necessidade de processos iterativos. O modelo de processo incremental combina elementos dos fluxos de processos tanto lineares quanto paralelos. A

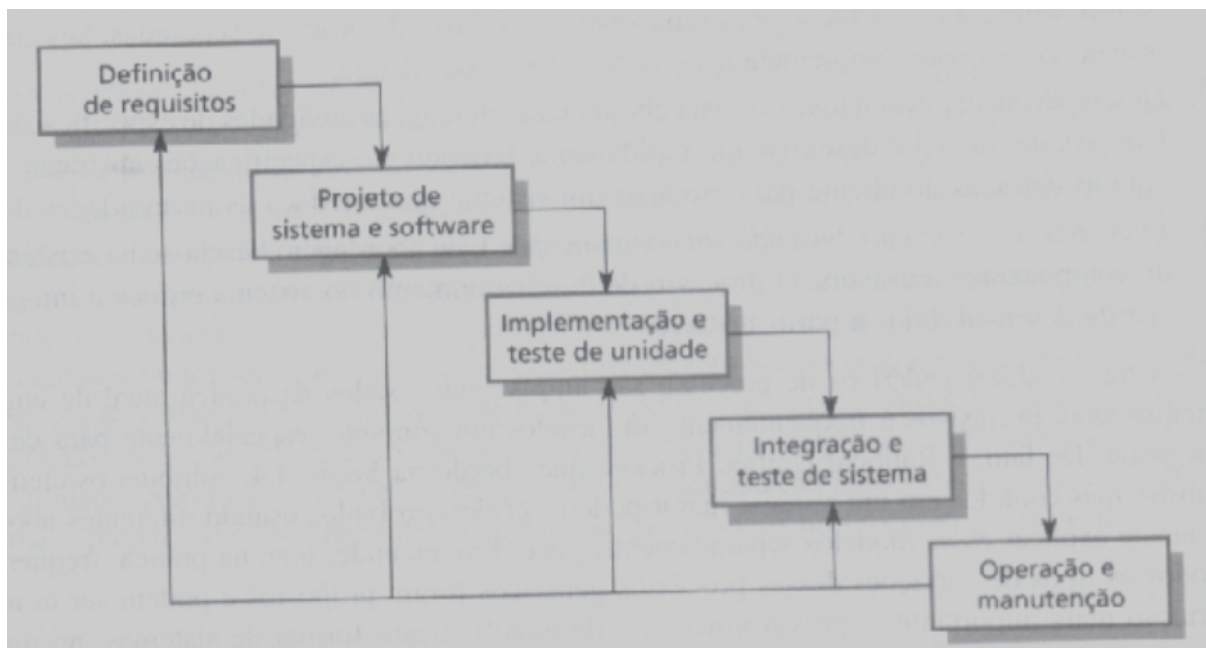


Figura 3.1: Modelo Cascata - Somerville (Somerville, 2007).

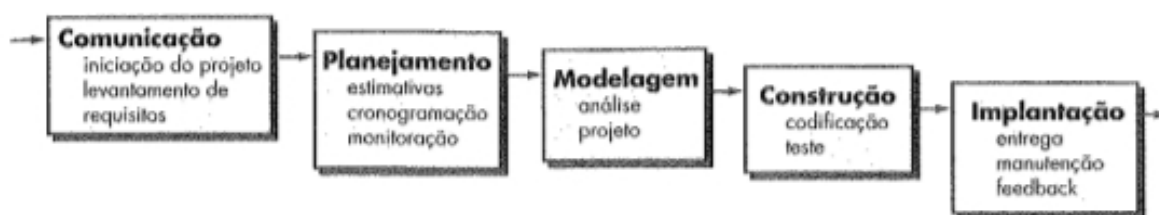


Figura 3.2: Modelo Cascata - Pressman (Pressman, 2010).

especificação, projeto e implementação do software são produzidos a cada incremento, um de cada vez. É um sistema bem documentado, contudo sempre sujeito a essas mudanças, ou seja, trata-se de uma abordagem intermediária. Como atesta *Somerville* através da figura 3.2 que representa este modelo:

No primeiro incremento de um produto que utiliza o modelo incremental tem-se o essencial do produto, ou seja, os requisitos básicos, de modo geral, que serão atendidos para o software entrar em operação e ser desenvolvido. Este primeiro incremento pode ser chamado de núcleo do produto, segundo Pressman (Pressman, 2010), e muitos recursos complementares ainda não são entregues para os clientes. Após a conclusão e entrega do núcleo do produto o cliente utiliza e avalia esse incremento através de uma revisão detalhada e fornecendo um resultado ou *feedback* de acordo com seu uso. Com base nesse resultado fornecido pelo cliente o próximo incremento é planejado considerando a modificação do primeiro incremento, caso seja necessário, de acordo com o *feedback do cliente*. Ou seja, cada incremento fornece um subconjunto para seu sucessor. A liberação de cada incremento é realizada através desse mesmo processo até que o produto esteja completo e de acordo com o que foi proposto e venha sendo testado pelo usuário. Para

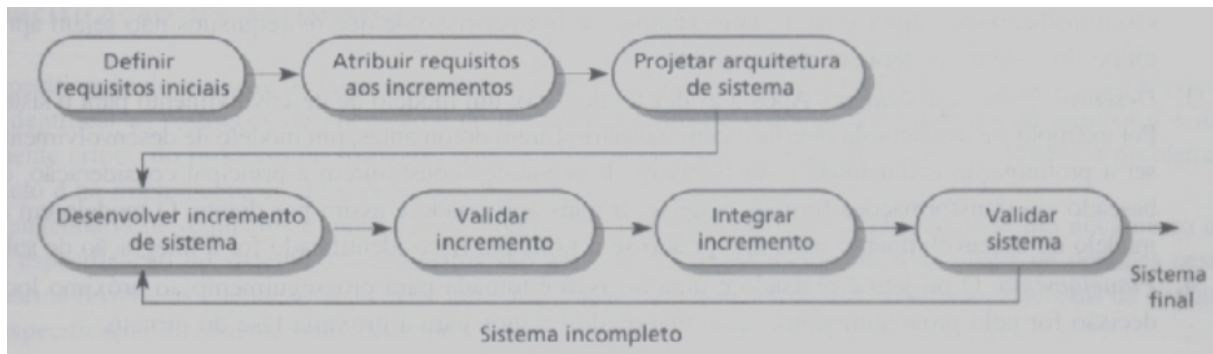


Figura 3.3: Modelo Incremental - Somerville (Somerville, 2007).

ilustrar este cenário temos a contribuição de *Pressman* com a figura 3.4:

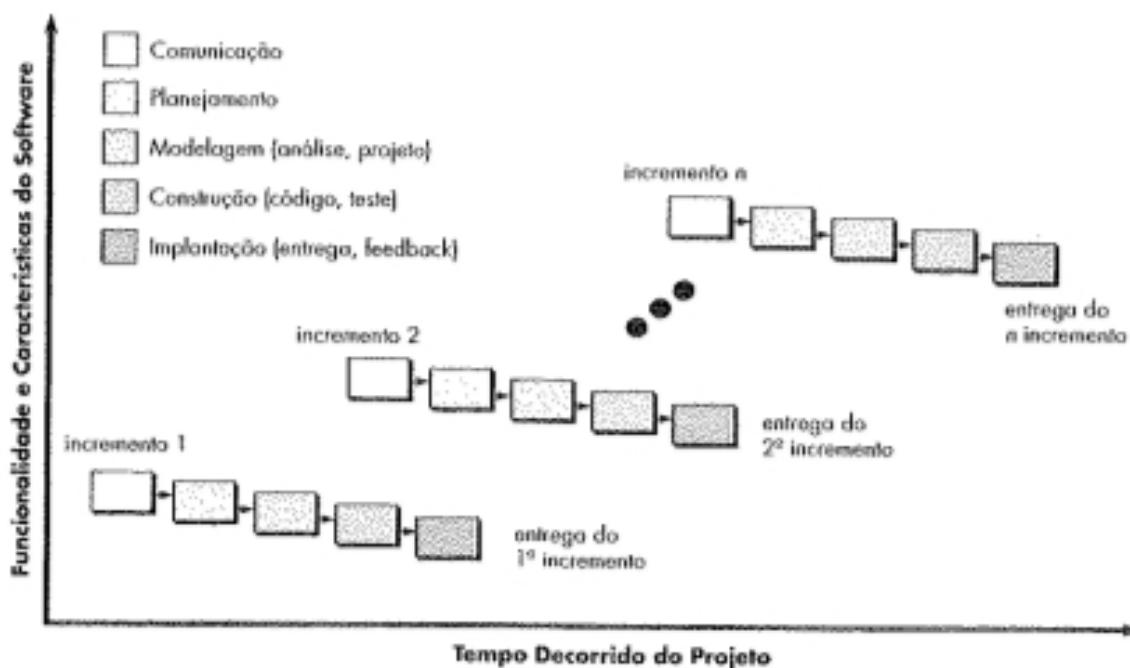


Figura 3.4: Modelo Incremental - Pressman (Pressman, 2010).

Esse modelo já traz algumas benfeitorias como: entrega um produto operacional a cada incremento; cada incremento pode funcionar sozinho, podendo e devendo ser integrados ao longo do desenvolvimento; redução de futuras falhas ao longo do processo, devida a constante entrega e *feedbacks* do cliente; geralmente as partes mais importantes do software ficam no início, e com isso passaram por mais testes, para adequação aos novos incrementos que serão adicionados. Doravante, têm-se o outro lado, uma possível dificuldade no mapeamento de cada incremento, vendo-se o tamanho foi adequado; e uma dificuldade na identificação quais poderiam ser os requisitos em comum para todos os outros incrementos.

3.2.3 Desenvolvimento em Espiral

Foi um modelo proposto pelo então cientista chefe da empresa *TRW Defense Systems Group*, Barry Boehm (Boehm, 1988). O modelo espiral é largamente utilizado e é considerado como uma abordagem realista para desenvolver sistemas em larga escala. Assim como o modelo incremental, o desenvolvimento em espiral, também se enquadra como modelo iterativo/evolucionário. O presente modelo foi pensado de forma a abranger outros modelos de processo. O processo não é uma sequência de atividades, ao contrário, é representado como uma espiral, e cada *loop* representará uma fase do processo (no primeiro à viabilidade, no segundo definição dos requisitos, projeto do sistema, e assim caminhando até o final do projeto). O caminho da espiral começa pelo centro e ao longo do processo são realizadas as atividades de cada quadrante, tomando por base Somerville (Somerville, 2007) e conforme ilustrado pela figura 3.5:

1. **Definição de objetivos:** proposição dos objetivos específicos, identificação das restrições sobre o processo e o produto visando preparação de uma plano de gerenciamento. E por fim a identificação dos riscos, para um possível planejamento de estratégias alternativas a ele. Este último ponto é o grande diferencial deste modelo;
2. **Avaliação e redução de riscos:** para cada risco identificado na fase anterior, será feita uma análise e tomada uma providência, buscando a redução dos riscos. No caso de desenvolver protótipos, caso os requisitos sejam inadequados;
3. **Desenvolvimento e validação:** escolha do modelo de desenvolvimento de acordo com o que foi avaliado na fase anterior, como riscos de interface do usuário, riscos de segurança, risco de integração de sistemas. Ou seja, escolhe se o modelo que melhor se adaptar ao risco verificado;
4. **Planejamento:** revisão do que foi feito e tomada de decisões para o próximo andamento da etapa no *loop* da espiral. Caso o projeto não tenha acabado, haverá o planejamento e análise para definição dos próximos passos;

Um ponto já destacado e que merece atenção nesse estudo é a consideração dos riscos neste modelo. Esses riscos são tratados como uma possibilidade de algo que possa acontecer de errado no processo de produção do software, como mudança da linguagem de programação, nova estrutura de um programa já implementado, mudança no banco de dados, entre outros. Os riscos podem acarretar: problemas no projeto, possibilidade de extensão do prazo proposto, e variação nos custos de produção do software. Com isso, tem se a importância de serem minimizados estes para um bom andamento do processo e entrega do software. Com a divisão do modelo, percebe se que após a verificação dos objetivos (desempenho e funcionalidade), são pensados caminhos alternativos visando seu alcance e a enumeração de cada restrição imposta é feita. Após a avaliação destas alternativas os riscos do projeto são então identificados. Pode se resolver através da coleta de informações, maior detalhamento da análise, e simulação, os riscos que surgirem. Para então prosseguir com as outras fases do modelo como a avaliação destes riscos no desenvolvimento para depois se planejar como será a próxima fase, ou ciclo da espiral.

Cada passagem pela parte de planejamento, por exemplo, resulta em ajustes do projeto. Os custos e o cronograma são sempre ajustados de acordo com o *feedback* obtido do

cliente após uma entrega. Também se tem um ajuste no número de interações planejadas para completar o software. Tem se a importância da interação com o cliente buscando o contínuo aperfeiçoamento do software proposto.

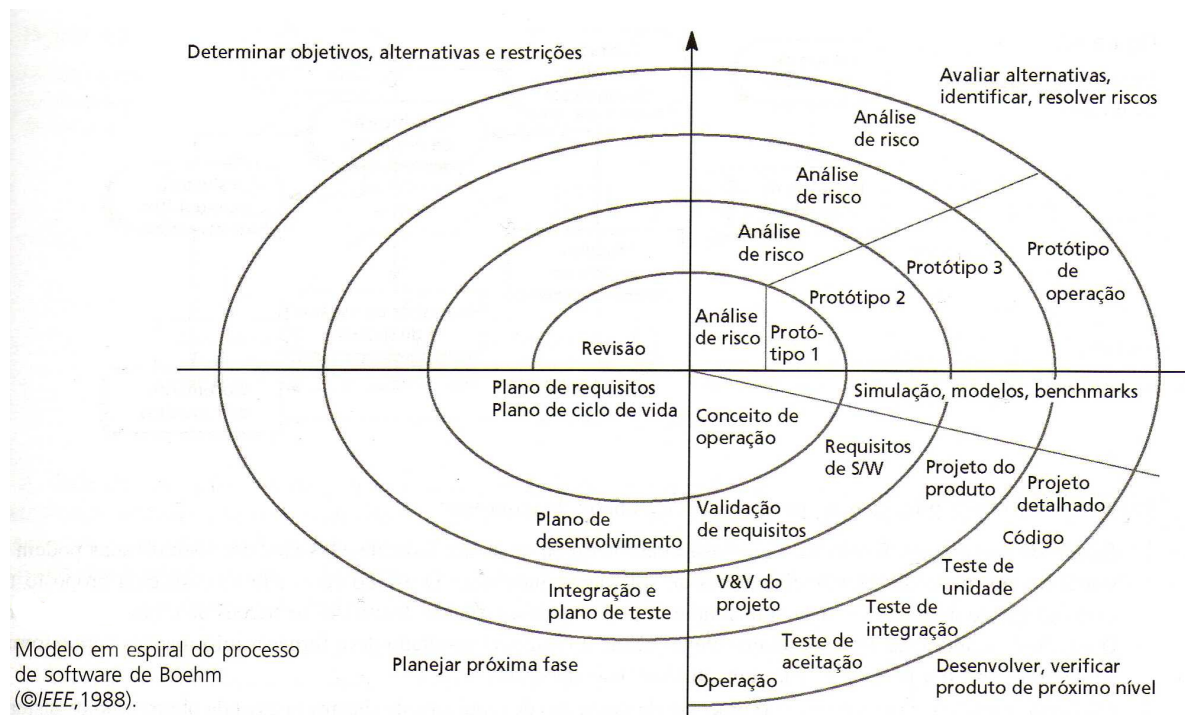


Figura 3.5: Desenvolvimento em Espiral - Somerville (Somerville, 2007).

3.2.4 Metodologia Ágil e o Modelo RUP

Atualmente, vive-se numa sociedade onde tudo é operado em escalada muitas vezes global e a cenários que exigem rápidas mudanças como mudança no mercado, de ordem econômica, novos produtos e serviços. O desenvolvimento e a entrega de softwares recebem especial atenção neste contexto. Há dificuldades de alguns clientes em definir requisitos estáveis, não sabem até a entrega do software como se comportará com outros programas interligados, ou seja, a experimentação do usuário é fundamental para avaliação destes requisitos e observação.

Daí surge uma nova concepção para o desenvolvimento de softwares nesse contexto, conforme aponta Somerville (Somerville, 2007). Vê-se a necessidade em se conceber processos de desenvolvimento rápido, que deve ter algumas características como: processos de especificação, projeto, e implementação concorrente, sem uma especificação detalhada do sistema. Ou seja, o usuário define o que de importante terá o sistema; desenvolvimento do sistema à base de incrementos, a cada incremento o usuário participa do processo - interação - podendo propor mudanças, novos requisitos, e essas modificações devem ser implementadas nos próximos incrementos que serão entregues; novas formas de interação com os usuários, que seria criação de novas plataformas, visando o acompanhamento do cliente.

Diante deste novo cenário, surge em 2001, o *Manifesto para Desenvolvimento Ágil de Software*. Elaborado por Kent Beck (Beck, 2001) e mais dezesseis integrantes, que declaram o seguinte:

“Estamos descobrindo maneiras melhores de desenvolver softwares, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano;

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.” Beck (Beck, 2001) - *Manifesto para Desenvolvimento Ágil de Software*

Este grupo de pensadores independentes sobre desenvolvimento de software, que se denominaram como *Aliança Ágil*, se baseiam em doze princípios para nortear o modelo ágil de desenvolvimento que são:

1. “Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;
3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
7. Software funcionando é a medida primária de progresso;
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade;
10. Simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial;
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis;
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo;”

Beck (Beck, 2001) - *Manifesto para Desenvolvimento Ágil de Software*

Pressman (Pressman, 2010), destaca que os principais pontos do pensamento ágil como um encorajamento de estruturas e atitudes de equipe, tornando a comunicação mais fácil entre seus membros e de todos os envolvidos no desenvolvimento do software. Tendo em vista que processo ágil prevê projeto e construção de software intercalados; a rápida entrega baseado na ideia de pequenos incrementos, contudo aponta que a menor importância dada a produtos de trabalho intermediários nem sempre é bom para o processo de produção; e a participação ativa dos clientes no desenvolvimento do projeto. Já sob o viés humano - um dos diferenciais desta metodologia ágil -, o autor destaca algumas características (Tabela: 3.1) para aqueles que venham a fazer parte de uma equipe que use essa metodologia, entre elas:

Ainda assim, Somerville (Somerville, 2007) afirma que há na literatura alguns questionamentos que apontam algumas imperfeições com relação ao modelo, entre elas: achar um cliente disposto a estar em contínuo contato com a equipe de desenvolvimento e os demais participantes envolvidos na produção do software. E uma pessoa não conseguiria representar todos os usuários que usufruirão do software; alguns membros das equipes de trabalho podem não interagir em harmonia com os outros colegas de trabalho; o fato de ser priorizado a mudança pelo modelo, pode gerar problemas quando ao invés de ter uma pessoa dialogando com as equipes, têm-se vários clientes num mesmo projeto. Neste caso, cada pessoa adotará prioridades diferentes para mudar algum ponto do programa; e por fim, nem sempre a simplificação do software será possível. Fatores como prazo curto para entrega final do sistema e membros não terem tempo suficiente para implementar as simplificações.

Como todos os outros, o método ágil, também tem seus limites e pontos positivos, conforme colocado no início deste capítulo. Tomando este pequeno estudo sobre a metodologia ágil, busca-se aqui nesta parte o estudo de um modelo de processo baseado neste tipo de metodologia.

O Rational Unified Process - RUP

Este modelo foi criado de maneira que engloba: alguns dos princípios da metodologia ágil; reconhece a comunicação com o cliente usando os casos de uso para descrever a visão do cliente; enfatiza o papel da arquitetura de software como compreensibilidade, abertura a modificações e reuso; seu fluxo de processo é iterativo e incremental; conforme nos coloca Pressman (Pressman, 2010). É um modelo derivado do trabalho sobre UML - Unified Modeling Language, é uma linguagem que define artefatos que ajudam a modelar e documentar os sistemas orientados a objeto - e é considerado como um modelo como um modelo híbrido de processo.

O principal objetivo do RUP é atender as necessidades dos usuários garantindo uma produção de software de alta qualidade, que cumpra um cronograma e um orçamento previsíveis. Somerville (Somerville, 2007) atesta que diferente dos modelos convencionais de processo de software que tem uma visão única do processo, o RUP traz três perspectivas:

- Dinâmica, mostra as fases do modelo ao longo do tempo;
- Estática, mostra as atividades realizadas no processo;
- Prática, sugestão de boas práticas a serem usadas no desenvolvimento do processo;

Características Pessoais numa Equipe Ágil	
Competência	<ul style="list-style-type: none"> - Talento Inato e Habilidades Específicas com relação ao software e ao processo; - Transmissão aos demais membros, habilidades e conhecimentos adquiridos
Foco Comum	<ul style="list-style-type: none"> - Integrantes focados na mesma meta; - Entrega de incrementos funcionando e dentro do prazo; - Adaptações contínuas para satisfazer às necessidades da equipe;
Colaboração	<ul style="list-style-type: none"> - Avaliar, analisar e usar as informações para a equipe de software; - Criar informações para ilustrar ao cliente o trabalho da equipe; - Construir informações para oferecer valor de negócio ao cliente;
Capacidade de tomada de decisão	<ul style="list-style-type: none"> - Autonomia para equipe sobre o andamento da produção; - Autoridade para tomada de decisões sobre partes técnicas e do projeto em si;
Resolver problemas vagos	<ul style="list-style-type: none"> - Reconhecer cenário de ambiguidades e modificações; - Pensar que o problema que está sendo resolvido, não aparecerá posteriormente; - Retirar um aprendizado da situação problemático;
Respeito e confiança	<ul style="list-style-type: none"> - Equipe que tenha confiança e respeito entre si, ajudando no fortalecimento do grupo como um todo;
Auto organização	<ul style="list-style-type: none"> - Organização para o trabalho ser feito; - Organizar o processo para um melhor acomodamento do seu ambiente local; - Organização do cronograma de trabalho, visando a melhor entrega do incremento do software para o cliente;

Tabela 3.1: Fatores humanos preponderantes numa equipe de trabalho usando metodologia ágil. Com modificações, Pressman (Pressman, 2010)

Este processo é centrado na arquitetura o que significa dizer que o conjunto de funcionalidades vai ditar a forma na qual o sistema será desenvolvido e como poderá ter manutenção. Além destas perspectivas apresentadas, tem-se as fases do RUP ao longo do processo de desenvolvimento de software. Estas quatro fases estão relacionadas mais aos negócios do que com a parte técnica do processo. As quatro fases são:

1. **Concepção:** primeira comunicação com o cliente, buscando desenvolver com cliente

e usuários finais o *business case* (requisitos do negócio) para o software. Atividades são: rascunho da arquitetura do sistema, um pequeno esboço dos subsistemas e funções do programa, que serão refinados ao longo do processo; desenvolvimento do plano interativo e incremental do programa; identificação de quem usará o software, como pessoas e outros sistemas, definição dessas relações através da elaboração dos primeiros casos de uso; avaliação do sistema, e estimativa do custo e do prazo, estabelecendo prioridades; e examinação dos objetivos do trabalho para decidir sobre sua viabilidade. O intuito dessa fase é criar uma primeira visão geral do projeto. E de 10% a 20% dos modelos de casos de uso são completados;

2. **Elaboração:** desenvolver um entendimento do problema através da modelagem do modelo genérico já elaborado, expansão dos casos de uso já elaborados (cerca de 80% a 90% do modelo já está criado), estabelecendo assim um *framework* para arquitetura do sistema com cinco visões diferentes: modelos de casos de uso, modelo de análise, modelo de projeto, modelo de implementação, e modelo de implantação. Há nesta fase também o reconhecimento dos riscos do projeto, e em alguns casos é elaborado uma primeira versão do sistema executável. Indagações como "O plano do projeto é confiável?", "Os custos são admissíveis?" são esclarecidas nesta etapa junto com o cliente. Finalizando esta fase tem-se os casos de uso da UML especificados que junto com os modelos já citados formam um modelo geral de requisitos do sistema, e a descrição da arquitetura, tendo assim um plano de desenvolvimento do sistema;
3. **Construção:** fase em que é desenvolvido o software tendo o modelo geral e sua composição (como os modelos de casos de uso), e o plano de desenvolvimento ambos gerados ao longo das duas fases anteriores do processo. Esta fase está relacionada diretamente com a elaboração, desenvolvendo a programação e os testes do sistema. Todas as partes são desenvolvidas paralelamente e é feita a integração delas nesta fase. Os modelos de análise e projeto são completados refletindo a versão final do incremento do software, sendo implementados no código fonte. Cada incremento tem sua fase de testes separados para depois serem integrados. Casos de uso são então usados nesse processo de integração, que já é a fase de testes de aceitação que são executados antes da próxima fase. Objetivo desta fase é que estes casos de uso sejam refletidos no software em produção;
4. **Transição:** fase final do modelo, onde o software que estava sendo desenvolvido é repassado ao cliente, fazendo o software entrar em funcionamento no seu ambiente real. Ou seja, o foco é assegurar que o software esteja disponível para seus usuários finais. É de fundamental importância o *feedback* do cliente para a pronta correção de defeitos e correções necessárias. A complexidade inerente nesta fase varia de acordo com o produto e cliente. Nesta fase podem ser definidas metas de acordo com o andamento do processo: se houver a alteração de uma falha de sistema pode envolver retrabalho na programação e teste, ou a correção de um requisito que possa levar a um processo similar ao de construção. Quando chega nesta fase tem-se o **Release do Produto (ou, PR – Product Release)**, a avaliação dos objetivos (se foram ou não atendidos), e na decisão de iniciar um novo processo de construção com base nesses objetivos. O software é entregue a documentação (como o material de treinamento, manual de instalação, e notas de release) e casos de uso que foram

usados, artefatos de instalação do mesmo, e é verificado o funcionamento correto do mesmo.

Este é o resumo que mostra como o sistema é construído usando o RUP. As fases dinâmicas do modelo são: definição do que será feito ao longo do projeto e quando deverão ser realizados, e a descrição das metas de desenvolvimento do projeto buscando o alcance do que o cliente solicitou.

Porém este modelo também tem uma visão estática priorizando as atividades durante o processo de desenvolvimento. Estas atividades também recebem o nome de *workflow*, também chamado de disciplina. Tais *workflows*, fazem parte dos artefatos (ou documentos) gerados ao longo das atividades, mostrando o que deve ser realizado em cada fase. Existem no RUP seis *workflows* de processo principais identificados e três *workflows* de apoio principais, que não são fixos, nem temporais nas fases acima descritas. Ou seja, um *workflow* pode ser usado em mais de uma fase, mesmo algum sendo usado prioritariamente naquela fase. A descrição destes é orientada pelos modelos UML associados a cada disciplina. Os primeiros seis são: modelagem de negócios, requisitos, análise e design, implementação, testes, implantação; e os outros de apoio são: gerenciamento de configuração e mudanças, gerenciamento de projetos e ambiente.

Os dados até então mostrados e sugeridos na literatura mostram que aplicações associam as disciplinas às fases. Mas como são tratadas como perspectivas dinâmicas e estáticas propõem justamente que não existam restrições nas suas combinações durante o projeto. A figura 3.6 mostra um resumo do que já foi colocado sobre o respectivo modelo:

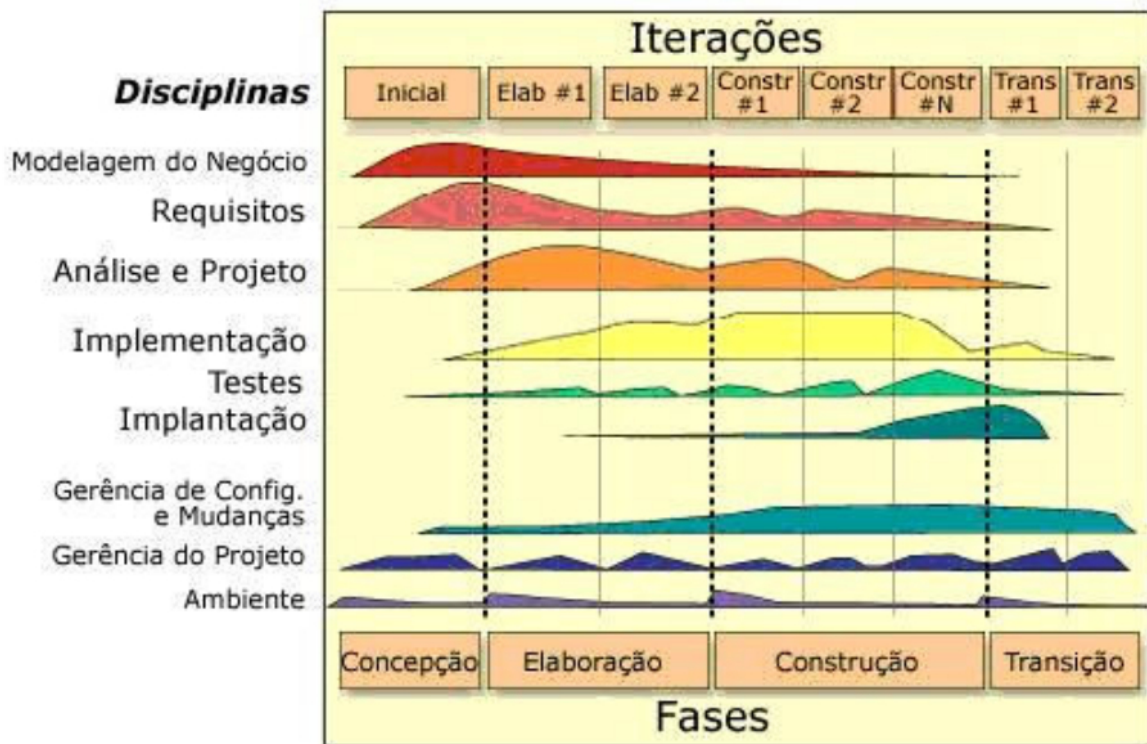


Figura 3.6: Visão Geral do RUP

Por fim o modelo traz, sob uma perspectiva prática, dicas recomendadas para o desenvolvimento do sistema, as seis práticas são:

1. *Desenvolvimento iterativo*: planejar, desenvolver e entregar os incrementos do sistema de acordo com o que foi priorizado com o cliente e de acordo com as características de maior prioridade no processo de desenvolvimento do programa;
2. *Gerenciar requisitos*: documentação formal dos requisitos, manter e analisar o impacto das mudanças no sistema antes de colocá las em prática;
3. *Arquiteturas baseadas em componentes*: estruturar a arquitetura de sistema em componentes, externos ou produzidos no projeto, facilitando a produção;
4. *Modelagem Visual*: utilização dos modelos UML representando o software de forma estática e dinâmica;
5. *Verificar a qualidade do software*: garantir através da monitoração e condução que no programa esteja sendo atendido os padrões de qualidade da organização, de aceitação do cliente, e nos padrões de alguma norma que o programa tenha que seguir;
6. *Controlar mudanças do software*: gerenciamento das mudanças durante o processo de produção do software, através de um sistema de gerenciamento de mudanças e procedimentos, ou ferramentas de gerenciamento de configuração. Ajudando na construção de um projeto melhor estruturado, documentado e aceito.

Por fim cabe ressaltar que neste trabalho foi considerado o RUP como um exemplo de metodologia ágil. Somerville (Somerville, 2007) traz o RUP como bom exemplo de modelo híbrido de processo também. Contudo alguns autores atualmente o consideram como um exemplo também de metodologia ágil por verificar nele alguns pontos do manifesto ágil como: preconização de equipes de trabalho no desenvolvimento, na produção dos incrementos paralelamente e na integração dos mesmos, nos remete à valorização maior do indivíduo e das interações por exemplo; mesmo possuindo uma documentação vasta, com casos de uso e UML's o RUP busca sempre a entrega do produto funcionando para o cliente; desenvolvimento iterativo no modelo, também é uma resposta rápida diante de alguma mudança no processo de produção; e o cliente tem papel fundamental nas primeiras fases do modelo. Ou seja, tem se uma clara influência do manifesto ágil sobre o RUP. Fowler (Fowler, 2005), ressalta que por ter como característica ser um *framework* que pode vir a ser adaptado, o RUP pode ser tratado tanto quanto mais prescritivo quanto ágil. No caso ágil devendo adaptá lo para uma forma mais “enxuta” no volume de atividades e artefatos.

Capítulo 4

Um modelo de valor do trabalho em software segundo a teoria do valor do trabalho

Neste capítulo apresenta-se uma proposição de modelo de processo de software baseado na teoria de valor do trabalho de Karl Marx. Este modelo tem por base o tripé formado primeiramente, por esta teoria, juntamente com o estudo sobre a metodologia ágil e o modelo de processo de software em espiral. O presente modelo busca permear tanto a área referente a sociologia do trabalho quando à engenharia de software, que com o presente estudo estão cada vez mais conjurando em certos aspectos.

O foco não é somente a produção e o desenvolvimento do software, mas sim uma tentativa de conciliar um modelo que trate do processo, mas que também acrescente o viés do trabalhador e o esforço que venha a ser empreendido na elaboração do software. Ao fim deste capítulo buscar-se-á um modelo tomando por base o que foi abordado neste estudo.

4.1 Princípios da teoria do valor do trabalho no contexto do desenvolvimento de software

Tomando por base os seguintes fatores já abordados neste estudo: a teoria do valor do trabalho na revolução informacional, a produção de software diretamente relacionada ao setor de serviços, a nova concepção de trabalho imaterial e sua função geradora de trabalho nesta produção. Pode-se resumir a teoria do valor do trabalho em software da seguinte forma: “o valor final de um software estará associado ao trabalho objetivado, gerador de valor em seu desenvolvimento, e através da cooperação entre atores internos (equipes de trabalho e gerentes) e externos (os clientes) que nele operaram e auxiliaram em sua concepção e elaboração.”

Da teoria do valor do trabalho percebe-se que o foco é o trabalhador e seu trabalho (atividade), ou seja, o trabalho é o gerador de valor (perde-se força do valor financeiro) para esta teoria. Partindo da esfera pessoal e perpassando até a social, onde conforme Marx traz, cada trabalhador com sua mercadoria específica forma um trabalho em sociedade. A atividade produtiva e o esforço despendido nesta ação geram valor. E bem

pode ser perceber esse mesmo raciocínio no campo do processo de desenvolvimento de software quando a metodologia ágil traz como premissa pessoas ao invés de processos, e vendo nos modelos de processos a importância do trabalho empreendido para a geração do produto: o software, por exemplo. A crítica que alguns autores trazem é que atualmente a relação de trabalho imaterial (homem - máquina) tem sido difícil a mensuração do esforço do trabalho. Lojkine (Lojkine, 1995) alerta que de um lado tem-se todo potencial dos sistemas tecnológicos e do outro lado a precarização e intensificação do trabalho. Porém o protagonismo da pessoa é preponderante na definição da teoria e neste estudo, tendo em vista que a figura do trabalhador tem ganho evidência na discussão sobre os processos de software.

O fator tempo ganha atenção e é ressaltado na teoria do valor do trabalho, trazendo que quanto mais tempo é empreendido (“gasto”) no processo de produção de mercadoria, mais valor essa terá. Fazendo o paralelo na área de produção de software, quanto mais tempo é despendido na elaboração de um software, mais se agrega valor ao produto, pelo fato do mesmo exigir uma maior complexidade em sua elaboração, ou maiores riscos. Porém o interessante é a mudança na concepção e como é empregado esse tempo, quando Marx define sua teoria o foco principal é a força física, na sociedade informacional cada vez mais a força se torna intelectual.

4.2 Apresentação do modelo de valor do trabalho em software

O presente modelo trará como base a teoria do valor do trabalho, a metodologia ágil e o modelo espiral. No que se refere a parte da sociologia já foi relatada no tópico anterior. Faltando então discorrer sobre os dois últimos assuntos.

Na metodologia ágil no que se refere à cooperação pode-se concluir conforme a tabela 3.1, que a pessoa que trabalha neste cenário informacional colabora com sua própria equipe, com aqueles que estão gerenciando o processo, e principalmente, com o cliente. Esse relacionamento é fundamental para o andamento do processo de software e sua evolução. Pode-se perceber que tem-se uma ótica voltada para o aspecto social, onde eu me relaciono com o outro que no caso é a figura do cliente e meu par da equipe de trabalho. Tem-se então que o tempo nesta metodologia possui um aspecto social também quando relacionado com estes atores apresentados.

Nesta metodologia o tempo relacionado a entrega e as pessoas são geradores de valor monetário e de sucesso no projeto. Isto é, quando menos tempo é gasto na entrega de um software maior valor econômico terá. Doravante para se chegar a essa maximização do tempo foi necessário também que as pessoas envolvidas no projeto maximizem seu tempo de trabalho. Quanto mais pessoas trabalharem num projeto durante um tempo maior, este levará menos tempo para ser concluído, tendo assim, um maior valor econômico. Daí tem-se um enlace entre a teoria do valor do trabalho e a metodologia ágil, tem-se uma valorização do trabalho de quem produz um software que se reflete no valor final da mercadoria.

No modelo de processo de software espiral, tem-se como um dos princípios a interação no processo de desenvolvimento de um programa entre os desenvolvedores e o cliente, a cada ciclo completado na espiral são realizados ajustes no plano do projeto, revisão

de custos e tempo como já foram colocados. As interações são o motor das iterações (repetição) do software, em cada passagem fica mais claro para quem desenvolve e para o cliente os riscos e como funcionará o programa. Há neste modelo um processo dinâmico, que irá basear o valor econômico do programa. Outro fator retirado do modelo em espiral é a questão da identificação dos riscos, onde no espiral eles são detectados e busca se pelas minimizações dos mesmos. Neste modelo usaremos a palavra criticidade, buscando verificar o quão crítico consistirá o incremento realizado.

Modelo de Valor do Trabalho em Software		
Dimensão	Descrição	Descritores
Tempo	tempo empregado na elaboração do software segundo o cliente, padrão e o trabalhador	Tempo previsto: tempo planejado e contratado
		Tempo realizado: tempo gasto pelo trabalhador
		Tempo esperado: tempo esperado com o cliente
Esforço do Trabalho	esforço do trabalho segundo o trabalhador e sua equipe de trabalho, onde cada incremento tem um valor crítico na composição geral do software	Esforço do trabalhador no que lhe foi atribuído realizar
		Esforço da equipe de trabalho para entrega de cada incremento
		Nível de Criticidade do Incremento quão crítico é aquele incremento na composição do software.
Interações	interações entre cliente e equipe de desenvolvimento no processo de elaboração do software	Quantidade de interações ao longo da produção do software, ao final dos incrementos entregues
		Resultado qualitativo das interações e outros feedback's, verificação do impacto no desenvolvimento do software.
Conhecimento	conhecimento empregado no software, podendo ser inovador ou aplicação de outro conhecimento já adquirido na elaboração do software	Grau de Inovação, análise se o que foi produzido é algo que não existia antes
		Grau de Aplicação, caso o trabalho não tenha sido inovador, qual aplicabilidade foi utilizada em sua elaboração
Valor Social	importância do software produzido em seu contexto social	Tempo social: tempo de entrega do software no mercado
		Valor qualitativo gerado pela entrega do software
		Conhecimento adquirido pela troca de experiências

Figura 4.1: Modelo de Trabalho em Software.

Cabe agora discorrer sobre o conjunto de atividades propostas por este modelo. Busca se nesta configuração atacar os seguintes pontos: o esforço do trabalho dentro do desenvolvimento de software, a inovação ou aplicabilidade que foram despendidos ao longo do processo, a importância das interações que acontecem entre cliente e desenvolvedores para construção do software. Tais fatores contribuem então para uma maior maximização do tempo do processo, gerando um maior valor social do trabalho realizado.

4.3 Métricas de Valor do Trabalho em Software

Esta mensuração proposta não objetiva uma “desqualificação” do trabalho dos atores que a produzem. Ao contrário busca se uma valorização do trabalho empregado para produzi lo. Tendo de antemão visto as dimensões e seus descritores, cabe agora apontar algumas propostas de métrica para o modelo apresentado.

Métrica de Valor do Trabalho em Software:	
Tempo	<ul style="list-style-type: none"> - Tempo proposto pelo cliente; - Tempo proposto pela equipe de desenvolvimento; - Tempo consensual entre cliente e equipe de desenvolvimento; - Tempo acrescentado ao tempo consensual; - Tempo total do projeto; - Tempo de cada incremento; - Tempo do esforço de cada funcionário;
Esforço do Trabalho:	<ul style="list-style-type: none"> - Criticidade de cada incremento; - Esforço total dos integrantes da equipe ao longo do desenvolvimento; - Esforço total do trabalhador na elaboração do projeto;
Interações:	<ul style="list-style-type: none"> - Interação inicial do cliente; - Interação consensual entre cliente e equipe de desenvolvimento; - Interação ao final de cada incremento entregue; - Interação final do software; - Feedback's do cliente;
Inovação e Aplicação:	<ul style="list-style-type: none"> - Coeficiente de inovação e aplicação cognitiva;
Valor Social:	<ul style="list-style-type: none"> - Tempo Social da Entrega do software; - Valor social do software e sua importância neste contexto; - Troca de experiência e conhecimento ao longo das interações;

Tabela 4.1: Métrica de Trabalho em Software

1. **Tempo:** dimensão proposta com o objetivo de mensurar o quanto de trabalho foi empregado com base nas particularidades de cada incremento. A expectativa do tempo para realização do software tanto pelo cliente, quanto pelo patrão que busca satisfação de seu cliente.

- *Tempo proposto pelo cliente:* tempo que o cliente apresenta na proposição do software. Geralmente se dá quando o cliente apresenta o que espera que o software venha a ter, e depois dá o prazo que necessita para que o software seja entregue e pronto para o uso;
- *Tempo proposto pela equipe de desenvolvimento:* depois da primeira reunião a equipe de trabalho e desenvolvimento, calcula o tempo que será necessário para elaboração do software. Já avaliando a criticidade dos incrementos, o tempo que cada incremento levará para ser desenvolvido de acordo com o quão crítico cada incremento será;
- *Tempo consensual entre cliente e equipe:* uma nova interação com o cliente acontece e nela a equipe de software apresenta o pré projeto do software e indica o tempo que será gasto. Caso esse tempo tenha divergência com o tempo

colocado pelo cliente, é feita uma adequação, ou em relação a esse tempo, ou ao software (podendo ter uma revisão do mesmo, quanto as funcionalidades, atributos, entre outros). De forma a adequar o que se deseja e em quanto tempo se espera que esteja pronto;

- *Tempo acrescentado ao tempo consensual:* ao longo do desenvolvimento do software, poderá se verificar uma provável extensão do tempo consensual. Ao final do incremento, onde será repassada ao cliente essa reavaliação do tempo de entrega final do software, terá uma nova interação para definição de quanto tempo será adicionado;
- *Tempo total do projeto:* será o tempo considerado como o prazo para elaboração, desenvolvimento e entrega do software ao cliente. Já sendo adicionado os possíveis acréscimos calculados, e acordados ao longo do desenvolvimento do software;
- *Tempo de cada incremento:* tempo final de cada parte já elaborada do projeto e entregue para o cliente. Fundamental para o acompanhamento de elaboração e conscientização tanto da equipe que vem desenvolvendo o programa, quando do cliente;
- *Tempo do esforço de cada funcionário:* tempo que cada funcionário trabalhou durante a elaboração e produção do software. Marco importante para se ter uma noção e acompanhamento do trabalhador e do chefe, que ilustra sob o viés do tempo o esforço dele (trabalhador) na produção do produto;

Com isso a dimensão do tempo tem tanto um viés econômico, ou seja, quanto entregue mais rápido o projeto maior valor econômico será atribuído com relação ao processo. E através desse modelo ganha um viés social, buscando valorizar o tempo de cada funcionário e o resultado da interação entre equipe e o cliente. O tempo do esforço é inversamente proporcional ao tempo final do projeto, dado que quanto mais o funcionário se dedicar a um processo, imagina se que um menor tempo será necessário para conclusão do mesmo. E uma relação diretamente proporcional entre tempo de cada incremento e tempo final, onde quando o tempo do incremento for reduzido, mais impacto terá na redução do tempo final;

2. **Esforço do Trabalho:** dimensão relacionada com o esforço do trabalhador ao longo do processo, tanto sob ótica pessoal, como da equipe de trabalho que atuou no desenvolvimento do software. É o ponto onde busca se a correlação com a crítica sobre a teoria do valor do trabalho no processo de produção de software.

- *Críticidade de cada incremento:* refere se a quão crítico é o incremento que esteja sendo implementado. Esta noção varia de um (no caso de um incremento crítico básico) a cinco (que seria um incremento com nível crítico alto, com uma maior dificuldade para implementar), ou seja, quanto mais elevado for o nível, maior dificuldade de implementação e, na maioria das vezes, maior será a importância desse incremento no software como um todo;
- *Esforço total dos integrantes da equipe ao longo do desenvolvimento:* somatório do item acima apurado de modo a ser usado tanto pelo gerente de projeto, para novos talentos para outros projetos, ou reavaliação de cargos. Também pode

ser usado ao longo do projeto pela equipe de trabalho, de modo a incentivar a maximização do tempo e adequação dos prazos estipulados;

- *Esforço total do trabalhador na elaboração do projeto:* importante ponto para a valorização do trabalho de cada pessoa da equipe. Ponto que busca estar em sintonia com o conceito de Marx, de quanto mais trabalho é empreendido maior valor esse produto terá. Ou seja, se o trabalhador atuou em incrementos com o item de criticidade alto, maior será o seu esforço despendido na elaboração do projeto;

Tem-se assim que esta dimensão busca trazer um viés mais crítico com relação aos trabalhadores e seu esforço na elaboração de software. Na tentativa de inserir esta crítica no cenário da engenharia e produção de software. Correlacionando com a teoria do valor do trabalho, busca-se mensurar o trabalho de cada integrante, do grupo e de cada incremento da produção do software;

3. **Interações:** esta dimensão busca valorar as interações entre cliente e equipe de desenvolvimento ao longo do processo de produção de software. Tal dimensão vem em consonância tanto com a metodologia ágil e com o processo em espiral, estreitando esses laços para que o produto seja desenvolvido conforme à necessidade do cliente.

- *Interação inicial do cliente:* marco em que o cliente apresenta a proposta do seu software e o que ele necessita;
- *Interação consensual entre cliente e equipe de desenvolvimento:* segundo marco onde a equipe de desenvolvimento analisa a proposta de software pelo cliente, e mostra ao mesmo a viabilidade do que foi proposto, do que pode ser feito para que o cliente reavalie o que foi proposto primeiramente. Ao final desta interação o software é dividido e repassado a equipe de desenvolvimento;
- *Interação ao final de cada incremento entregue:* ao final de cada incremento (ou módulo) de software é apresentado ao cliente já para seu uso. Com isso o cliente tem a possibilidade de já fazer uso do software e intervir caso necessário;
- *Interação final do software:* ao final do processo de produção do software é feita essa interação de forma a avaliar se o que foi definido inicialmente foi entregue e se o software está compatível
- *Feedback's do cliente:* marco visando outras contribuições do cliente ao longo do processo, além das interações a cada entrega;

4. **Inovação e Aplicação:** relacionado com a dimensão do Conhecimento no Modelo de trabalho em Software, que busca agregar valor ao tempo de esforço do trabalho, considerando o quanto este trabalho tem de inovação ou aplicação de conhecimentos já adquiridos.

- *Coefficiente de inovação e aplicação cognitiva:* coeficiente que será usado para avaliar o grau de inovação e aplicação usado na elaboração do software. Tomando a mesma noção de valoração adotada no item de criticidade de cada incremento: tomando o nível um (no caso de um nível de aplicação ou inovação básico) a cinco (que seria um nível alto, com uma maior elaboração e até mesmo uma nova implementação na elaboração do software);

Cabe ressaltar que para uma mensuração desse item, deverá ser revisto todo o software na busca de comparações entre os incrementos que foram elaborados, para que assim haja essa diferenciação entre os incrementos feitos e/ou criados ao longo da elaboração do projeto de software.

5. **Valor Social:** trabalha a contribuição do software para a sociedade de forma geral, ampla. Nesta dimensão não há pontos quantitativos, ainda assim, busca se a ênfase na verificação qualitativa do software produzido.

- *Tempo Social da Entrega do software:* busca importância da entrega do software no prazo acordado com o cliente, a credibilidade da empresa responsável pela entrega e produção do software, competência da equipe que trabalhou no projeto;
- *Valor social do software e sua importância neste contexto:* qual importância e uso do software produzido para sociedade ou na realidade que foi pensado sua elaboração, qual mudança que o uso do software produzirá;
- *Troca de experiência e conhecimento ao longo das interações:* visa ressaltar a importância dos conhecimentos que foram adquiridos entre os atores no processo principalmente a relação entre o cliente e os gerentes responsáveis pela elaboração do mesmo;

Pode se observar que as dimensões propostas pelo modelo (tempo, esforço do trabalho, interações, conhecimento e valor social) resultam da articulação das áreas do conhecimento abordadas. Tendo a métrica do trabalho sendo formulada a partir dos valores sociais entre os atores envolvidos no processo de produção do software (cliente, patrão e trabalhador), revisão constante das entregas que forem realizadas ao longo do processo, detalhamento e revisão de cada parte do programa, valorização de quem participou do trabalho de construção do software através do tempo, e por fim, analisou o valor social que o software produzido terá ao ser entregue para o cliente final. Percebe-se que as dimensões propostas estão em consonância com o que foi estudado ao longo do trabalho, sendo trabalhadas de formas interligadas propondo uma nova forma de se pensar o trabalho ao longo do processo de software.

A dimensão do valor social e de esforço do trabalho propostas serão os pontos que merecerão maior cuidado quanto à aplicação. Por se tratarem de dimensões mais ligadas à área da sociologia do trabalho, podem causar num primeiro momento uma certa dificuldade para implementação ao longo do processo de produção. Gerando o desafio de observar e valorizar o trabalhador ao longo do processo de produção e ressaltando o valor gerado pela entrega do produto ao cliente. Daí tem-se a inovação deste trabalho e o que ele propõe de novo diante dos modelos já utilizados no mercado.

Capítulo 5

Considerações Finais

O setor de serviços na sociedade informacional tem obtido crescimento sólido tanto em arrecadação quanto na parcela de atuação no mercado de Tecnologia da Informação. Contudo, na perspectiva do trabalhador, esse crescimento não tem sido representado nos dados objetivos sobre o esforço de trabalho nesse setor. Observamos a falta de compreensão e consenso sobre os componentes de medida e de valor do trabalho no processo de criação e produção imaterial dos produtos e serviços inerentes ao mercado na sociedade informacional.

Nesse contexto, o estudo envolveu as áreas interdisciplinares da Sociologia do Trabalho e da Engenharia de Software para aprofundar o entendimento sobre o valor do trabalho imaterial no processo de desenvolvimento de software. Os pressupostos da teoria crítica do valor do trabalho e do trabalho imaterial no contexto da sociedade da informação e do conhecimento foram abordados. Na perspectiva da sociedade informacional e da teoria crítica do valor do trabalho, encontramos no trabalho imaterial ou de criação e produção de bens intangíveis a raiz do problema. Essa questão, como ponto de partida, nos ajuda a refletir primeiramente sobre a questão conceitual envolvida no trabalho de produção de software para subsidiar as decisões de valor na composição do preço de um produto ou serviço imaterial ou de software.

O método exploratório e de pesquisa documental subsidiou o estudo a partir dos conceitos da teoria do valor do trabalho e do trabalho imaterial; e dos modelos de engenharia de software em cascata, incremental, de espiral e de desenvolvimento ágil. Tais fundamentos subsidiaram o conceito de valor final de um software associado ao trabalho objetivado, gerador de valor em seu desenvolvimento, pela cooperação entre atores internos, das equipes de trabalho e gerentes, e externos, os clientes, que operam juntos e auxiliam na concepção e elaboração do produto ou serviço.

O estudo resultou na proposição de um Modelo de Valor do Trabalho em Software com cinco dimensões as quais incluem: a) dimensão do tempo; b) dimensão do esforço do trabalho; c) dimensão das interações; d) dimensão do conhecimento; e) dimensão do valor social. Essas dimensões visam subsidiar métricas abrangentes de valor do trabalho no desenvolvimento de software que consideram além dos parâmetros de esforço, de tempo e de inovação, os aspectos de interações que ocorrem entre o cliente e a equipe de trabalho na construção do software. As dimensões propostas contribuem para explicitar o valor social do trabalho realizado segundo um conjunto de métricas que poderão ser aplicadas no desenvolvimento e produção de software pelas empresas do setor.

O referido modelo não constitui métrica específica e, como um modelo geral de referência, poderá subsidiar estudos de aperfeiçoamento e aplicação no mercado e no desenvolvimento de software nas organizações. Na construção do modelo alguns pontos da alienação do trabalho no setor de serviços foram evidenciados para iluminar uma proposta preliminar frente aos modelos de processo de software existentes. Por esses aspectos, o estudo e o modelo poderão contribuir para avanços nas empresas do setor de produção e de mercado em software. Na área da sociologia do trabalho, estudos posteriores poderão avançar na análise e adequação do modelo pois trata-se de um recorte que sintetiza a discussão do homem e o seu papel na sociedade, ou seja, da dimensão do indivíduo confrontada no seu contexto social.

Nessa perspectiva, o trabalho traz novos horizontes para modelos e métricas computacionais do trabalho imaterial de desenvolvimento de software sustentada pela sociologia do trabalho na perspectiva crítica. Como sugestão de trabalhos futuros, pode ser considerada a aplicação do modelo nas empresas de software, no ramo de serviços e em situações cotidianas, para revisão do modelo, de suas dimensões e métricas propostas, bem como de sua aplicação e validação prática.

Referências

- ABES, Associação Brasileira das Empresas de Software. 2014. *Mercado Brasileiro de Software - Panoramas e Tendências*. <http://central.abessoftware.com.br>. [Acessado em: 03 de Setembro de 2015.]. vi, 1, 13, 14, 15
- Antunes, Ricardo. 2008. Desenhando a nova morfologia do trabalho: As múltiplas formas de degradação do trabalho. *Revista Crítica de Ciências Sociais [Online]*, 83, 19–34. [Disponível em: <http://rccs.revues.org/431> - Acessado: 25 de Agosto de 2015.]. 1, 11
- Antunes, Ricardo, & Alves, Giovanni. 2004. As mutações no mundo do trabalho, na era da mundialização do capital. *Educação e Sociedade*, 25, 335–351. 13, 16
- Beck, Kent. 2001. *Manifesto para Desenvolvimento Ágil de Software*. <http://agilemanifesto.org/iso/ptbr/manifesto.html>. [Acessado em: 19 de Agosto de 2015.]. 26
- Boehm, Barry W. 1988. A Spiral Model of Software Development and Enhancement. *IEEE*. http://portal.ou.nl/documents/114964/2986739/T24331_02.pdf [Acessado em: 18 de Agosto de 2015.]. 24
- Fowler, Martin. 2005. *The New Methodology*. <http://www.martinfowler.com/articles/newMethodology.html>. [Acessado em: 25 de Agosto de 2015.]. 31
- Group, The Standish. 2010. *Chaos Report*. <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>. [Acessado em: 06 de Agosto de 2015.]. 19
- Group, The Standish. 2015. *Sobre o The Standish Group*. <http://brazil.standishgroup.com/index.php?r=site/page&view=about>. [Acessado em: 06 de Agosto de 2015.]. 19
- Lojkine, Jean. 1995. *A Revolução Informacional*. 2nd edn. Rua Bartira, 387, São Paulo, Brasil: Cortez Editora. vi, vii, 8, 9, 10, 11, 33
- Marx, Karl. 1996. *O Capital*. 1st edn. Rua Paes Leme, 524 - 10º andar CEP 05424-010 - São Paulo, Brasil: Editora Nova Cultural Ltda (Os Economistas). 5, 6, 7, 11
- Marx, Karl. 2011. *Grundrisse*. 1st edn. Rua Pereira Leite, 373 – Sumarezinho – CEP: 05442-000 - São Paulo, Brasil: Boitempo Editorial. 7
- Mattei, Lauro. 2003. Teoria do valor-trabalho: do ideário clássico aos postulados marxistas. *Ensaio FEE*, 1(1), 271–294. 3, 4, 5, 6, 7

- Pressman, Roges S. 2010. *Engenharia de Software*. 6nd edn. Av Jeronimo de Ornelas, 670, Santana, Porto Alegre, Brasil: Bookman. vi, vii, 17, 18, 19, 20, 21, 22, 23, 27, 28
- Ricardo, David. 1996. *Princípios de Economia Política e Tributação*. 1st edn. Rua Paes Leme, 524 - 10º andar CEP 05424-010 - São Paulo, Brasil: Editora Nova Cultural Ltda (Os Economistas). 4, 5
- Rosso, Sadi Dal. 2014. Teoria do valor e trabalho produtivo no setor de serviços. *Caderno CRH, volume 21*, 75–89. 12, 13
- Smith, Adam. 1996. *A Riqueza das Nações*. 1st edn. Rua Paes Leme, 524 - 10º andar CEP 05424-010 - São Paulo, Brasil: Editora Nova Cultural Ltda (Os Economistas). 4
- Somerville, Ian. 2007. *Engenharia de Software*. 6nd edn. Av. Ermano Marchetti, 1435, Lapa, São Paulo, Brasil: Addison Wesley. vi, 17, 20, 21, 22, 23, 24, 25, 27, 31